

## DAFTAR PUSTAKA

- [1] R. Jamal, A. A. Ikhval, N. A. Nisa, S. H. Qulbi, and M. U. Arifin, "Penggunaan Teknologi Informasi dalam Mengoptimalkan Supply Chain Management," *J. Inov. Glob.*, vol. 2, no. 7, pp. 737–750, 2024, doi: 10.58344/jig.v2i7.1117.
- [2] Z. Niqotaini, *Konsep Sistem Informasi*, vol. 3, no. 1. Yogyakarta: PT Penamuda Media, 2023. doi: 10.17509/jap.v3i1.6095.
- [3] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *J. Ilmu-Ilmu Inform. dan Manaj. STMIK*, no. Oktober, 2020.
- [4] F. Soufitri, *Konsep Sistem Informasi*, vol. 3. Sidempuan: PT Inovasi Pratama Internasional, 2023. [Online]. Available: <https://ejournal.upi.edu/index.php/JAPSPs/article/viewFile/6095/4116>
- [5] Irwanto, "Perancangan Sistem Informasi Sekolah Kejuruan dengan Menggunakan Metode Waterfall (Studi Kasus SMK PGRI 1 Kota Serang-Banten) Irwanto," *Lect. J. Pendidik.*, vol. 12, no. 1, pp. 86–107, 2021.
- [6] Z. Latuconsina and N. Sariwating, "Pengaruh Dimensi Dari Supply Chain Management Terhadap Kinerja Operasional Toko Komputer Di Kota Ambon," *J. Cita Ekon.*, vol. XIV, no. 2, pp. 67–80, 2020.
- [7] S. A. Lubis, A. Mahfadillah, and S. Aisyah, "Analisis Penerapan Supply Chain Management Pada Produk Dodol Pasar Bengkel Diperbaungan Serdang Bedagai," *J. Manaj. Akunt.*, vol. 2, no. 4, pp. 550–555, 2022.
- [8] M. Jamaludin, "Analisis Perencanaan Supply Chain Management (Scm) Pada Pt. Xyz Bandung Jawa Barat," *Kebijak. J. Ilmu Adm.*, vol. 13, no. 2, pp. 70–83, 2022.
- [9] Agnes Fitriani, Kelvin Kwek, Lydia Then, and Supriyadi Arifin, "Analisis Penerapan Erp Dan Scm Pada Pt Indofood Sukses Makmur Tbk," *J. Inov. Penelit.*, vol. 3, no. 1, pp. 4403–4414, 2022.
- [10] M. M. Fariq, I. Nur, Z. D. Nur, and A. Yusuf, "Supply Chain Management Model for Agricultural Industrial Products and Derivatives," *J. Chem. Inf. Model.*, vol. 6, no. 1, pp. 2443–3548, 2020.
- [11] S. Monalisa and D. Apsyarin, "Rancang Bangun Sistem Informasi Supply Chain Management Distribusi Barang Dan Jasa Berbasis Web," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 7, no. 2, pp. 138–144, 2021.
- [12] C. D. Hardiana and Nanda Setiawan, "Pengaruh Supply Chain Management (SCM), Manajemen Kualitas dan Strategi Inovasi Terhadap Kinerja Operasional pada Perusahaan Manufaktur di Kawasan Delta Silicon Cikarang," *J. Bus. Theory Pract.*, vol. 1, no. 2, 2021.
- [13] E. Awaliya *et al.*, "Analisis Pengaruh Penerapan Sistem Manajemen Enterprise Resource Planning (Erp) Dalam Meningkatkan Supply Chain Management (Scm) Pada Perusahaan Indofood Tbk," *J. Mirai Manag.*, vol.

8, no. 1, pp. 520–530, 2023.

- [14] H. Maret Wijaya, G. Deswanto, and R. Hidayat, “Analisis Perencanaan Supply Chain Management (Scm) Pada Pt. Kylo Kopi Indonesia,” *J. Ekon. Manaj. Sist. Inf.*, vol. 2, no. 6, pp. 795–806, 2021.
- [15] R. Islamia, I. R. Al Faizy, A. Aqilla, R. F. Ahmad, A. Z. P. Arum, and G. Pratama, “DAMPAK KENAIKAN HARGA BAHAN BAKAR MINYAK (BBM) TERHADAP SEMBILAN BAHAN POKOK (SEMBAKO) DI TOKO SANI KABUPATEN CIREBON,” *J. Ekon. Manaj.*, vol. 17, no. 2, pp. 1–7, 2022, [Online]. Available: <http://oaj.stiecirebon.ac.id/index.php/jem>
- [16] Siska Narulita, Ahmad Nugroho, and M. Zakki Abdillah, “Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS),” *Bridg. J. Publ. Sist. Inf. dan Telekomun.*, vol. 2, no. 3, pp. 244–256, 2024.
- [17] Rizka Hafsa, E. Aribi, and N. Maulana, “Perancangan Sistem Informasi Manajemen Inventori dan Penjualan pada Perusahaan PT. Inhutani V,” *PROSISKO*, vol. 10, no. 2, pp. 109–116, 2023.
- [18] U. Pratiwi, K. Wijaya, and F. Fajriyah, “Penerapan Metode Prototype Pada Perancangan Sistem Administrasi Pembayaran Karate Berbasis Website: Studi Kasus Lemkari Prabumulih,” *J. Pengemb. Sist. Inf. dan Inform.*, vol. 2, no. 3, pp. 157–173, 2021.
- [19] A. Muhammad, N. Aditya, and H. Abduh, “Sistem Informasi Akademik Sekolah Berbasis Web Pada Sdn 32 Lagaligo Palopo,” *Indones. J. Educ. Humanit.*, vol. 2, no. 3, pp. 93–107, 2022.
- [20] I. Budiman, S. Saori, R. N. Anwar, Fitriani, and M. Y. Pangestu, “Analisis Pengendalian Mutu di Bidang Industri Makanan,” *J. Inov. Penelit.*, vol. 1, no. 10, pp. 1–15, 2021.
- [21] Masrizal, R. Hidayatullah, and M. Arif, “Sistem Informasi Pengolahan Data Alumni Dan Tracer Study Pada STMIK DUMAI,” *Lentera Dumai*, vol. 12, no. 1, pp. 19–27, 2021.
- [22] A. P. A. Kurniawan, S. Achmadi, and A. Mahmudi, “Sistem Pendukung Keputusan Dengan Metode Simple Additive Weighting (Saw) Dalam Memilih Saham Badan Usaha Milik Negara (Bumn) Berbasis Web,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 1, pp. 301–308, 2022.
- [23] I. Marzuki, D. Ariyanti, and D. Firmansyah, “Aplikasi Sebaran Pengangguran di Kota Probolinggo Berbasis Web,” *J. ENERGY (Jurnal Ilm. Ilmu-Ilmu Tek.)*, vol. 10, no. 2, pp. 83–91, 2020, [Online]. Available: <https://ejournal.upm.ac.id/index.php/energy/>
- [24] S. P. Balqis, T. Ardiansyah, A. T. Al, and A. Siahaan, “Perancangan Sistem Penerimaan Siswa Baru SMK Swasta Yapim Biru-Biru Berbasis Web,” *J. Ris. Sist. Inf. dan Tek. Inform.*, vol. 2, no. 3, pp. 28–36, 2024, doi: <https://doi.org/10.61132/mercurius.v2i3.103>.
- [25] K. S. Ningsih, N. J. Aruan, and A. T. A. A. Siahaan, “Aplikasi Buku Tamu

- Menggunakan Fitur Kamera dan Ajax Berbasis Website pada Kantor Dispora Kota Medan,” *J. Sains, Inform. dan Teknol.*, vol. 8, no. 1, pp. 1–6, 2023.
- [26] R. Hermiati, A. Asnawati, and I. Kanedi, “Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database Mysql,” *J. Media Infotama*, vol. 17, no. 1, pp. 54–66, 2021.
- [27] J. R. Paritama, “ANALISIS PERBAIKAN KINERJA RANTAI PASOK PADA INDUSTRI GARMEN DENGAN METODE SUPPLY CHAIN OPERATION REFERENCE DAN ANALYTICCAL HIERARCHY PROCESS (SCOR-AHP),” *J. Tek. Ind.*, no. April 2023, pp. 1–5, 2023.
- [28] R. Andreana and Marwan, “Pengukuran Kinerja Supply Chain Management Di Toko Aki Menggunakan Metode SCOR,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 7, no. 2, pp. 245–252, 2025.
- [29] A. Nasir and E. Supriatna, “Pengaruh praktek Supply Chain Management (SCM) dan integrasi supply chain terhadap kinerja Perusahaan Pada PT. Indofood CPB Sukses Makmur Tbk,” *J. Innov. Res. dan Knowl.*, vol. 2, no. 3, pp. 621–628, 2022.
- [30] A. F. Yani, D. Artanti, and T. N. Wiyatno, “Analisis Efektivitas Supply Chain Management Meningkatkan Efisiensi Pasokan Pada PT Chemi-Con Indonesia,” *Indones. J. Econ. Strateg. Manag.*, vol. 2, no. 4, pp. 3023–3031, 2024.
- [31] A. R. B. Alam, Y. P. Negoro, and Hidayat, “PENGUKURAN KINERJA SUPPLY CHAIN DENGAN MENGGUNAKAN METODE SCOR ( Studi Kasus : Pabrik penggilingan padi di UD . Sumber Tani ),” *JUSTI (Jurnal Sist. Dan Tek. Ind.)*, vol. 5, no. 2, pp. 141–150, 2024.

## LISTING PROGRAM

### 1. Owner.php

```
<?php
defined('BASEPATH') OR exit('No direct script
access allowed');
```

```
class Owner extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load-
>model('Owner_model');
        if($this->session-
>userdata('status') != 'login_super') {
            redirect("");
        }
        public function dashboard() {
            $data = array (
                'title'
=> 'Dashboard',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'kategori'
=> $this->db-
>get_where('tb_kategori',['kat_user' =>
'62ee698ac485341c794c9cc5ff138272'])-
>num_rows(),
                'barang'
=> $this->db-
>get_where('tb_barang',['barang_user' =>
'62ee698ac485341c794c9cc5ff138272'])-
>num_rows(),
                'supplier'
=> $this->db-
>get_where('tb_barang',['barang_user' =>
'62ee698ac485341c794c9cc5ff138272'])-
>num_rows(),
                'penjualan'
=> $this->db->get('tb_transaksi')-
>num_rows(),
                'grafikpenjualan'=>
                $this->Owner_model-
>grafik_penjualan(),
                'grafikkategori'=>
```

```
                $this->Owner_model-
>grafik_kategori(),
            );
            $this->load-
>view('owner/header', $data);
            $this->load-
>view('owner/index', $data);
            $this->load-
>view('owner/footer');
        }
        public function kategori() {
            $data = array (
                'title'
=> 'Kategori',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'kategori'
=> $this->Owner_model-
>data_kategori(),
            );
            $this->load-
>view('owner/header', $data);
            $this->load-
>view('owner/kategori', $data);
            $this->load-
>view('owner/footer');
        }
        public function barang() {
            $data = array (
                'title'
=> 'Barang',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'barang'
=> $this->Owner_model-
>data_barang(),
            );
            $this->load-
>view('owner/header', $data);
            $this->load-
>view('owner/barang', $data);
            $this->load-
>view('owner/footer');
        }
        public function supplier() {
            $data = array (
```

```

                'title'
=> 'Supplier',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>->row_array(),
                'supplier'
=> $this->Owner_model-
>data_supplier(),
                );
                $this->load-
>view('owner/header', $data);
                $this->load-
>view('owner/supplier', $data);
                $this->load-
>view('owner/footer');
        }
        public function penjualan() {
            $data = array (
                'title'
=> 'Penjualan',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>->row_array(),
                'transaksi'
=> $this->Owner_model-
>data_transaksi(),
                );
                $this->load-
>view('owner/header', $data);
                $this->load-
>view('owner/transaksi', $data);
                $this->load-
>view('owner/footer');
        }
        public function stok_barang() {
            $data = array (
                'title'
=> 'Stok Barang',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>->row_array(),
                'barang'
=> $this->Owner_model-
>stok_barang(),

```

```

                );
                $this->load-
>view('owner/header', $data);
                $this->load-
>view('owner/stok_barang', $data);
                $this->load-
>view('owner/footer');
        }
        public function list_pemesanan() {
            $data = array (
                'title'
=> 'Daftar Pemesanan',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>->row_array(),
                'pemesanan'
=> $this->Owner_model-
>data_pemesanan(),
                );
                $this->load-
>view('owner/header', $data);
                $this->load-
>view('owner/pemesanan', $data);
                $this->load-
>view('owner/footer');
        }
        public function lap_kategori() {
            $data = array (
                'title'
=> 'Laporan Data Kategori',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>->row_array(),
                'kategori'
=> $this->Owner_model-
>data_kategori(),
                );
                $this->load-
>view('owner/header', $data);
                $this->load-
>view('owner/lap_kategori', $data);
                $this->load-
>view('owner/footer');
        }
        public function lap_kategori_print() {
            $data = array (
                'title'
=> 'Laporan Data Kategori',

```

```

        'me'
        => $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        'atr'
        => $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
        'kategori'
        => $this->Owner_model-
>data_kategori(),
    );
    $this->load-
>view('owner/lap_kategori_print', $data);
    }
    public function lap_kategori_excel() {
        $data = array (
            'title'
            => 'Laporan Data Kategori',
            'me'
            => $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'kategori'
            => $this->Owner_model-
>data_kategori(),
        );
        $this->load-
>view('owner/lap_kategori_excel', $data);
    }

    public function lap_kategori_pdf() {
        $title = 'Laporan Data
Kategori';
        $atr = $this->db-
>get_where('tb_atribut',['id' => 1])>row_array();
        $kategori = $this-
>Owner_model->data_kategori();
        $pdf = new
FPDF('p','mm','Legal');
        // membuat halaman baru
        $pdf->AddPage();
        // setting jenis font yang akan digunakan
        $pdf->SetFillColor(112, 226, 137);
        $pdf-
>Image(base_url()).'assets/admin/images/'. $atr['fa
vicon'], 30, 12, -300);
        $pdf->SetFont('Arial','B',16);
        // mencetak string
        $pdf->Cell(195,7,$title,0,1,'C');
        $pdf->SetFont('Arial','B',14);
        $pdf->Cell(195,7,$atr['nama'],0,1,'C');
        $pdf->SetFont('Arial','B',12);
        $pdf->Cell(195,7,$atr['alamat'],0,1,'C');

```

```

        $pdf->Ln(6);
        $pdf->SetFont('Times','B',11);
        $pdf->Cell(10,8,'No',1,0,'C',1);
        $pdf->Cell(96,8,'Kategori',1,0,'C',1);
        $pdf->Cell(90,8,'Jumlah Barang',1,1,'C',1);
        $pdf->SetFont('Times','',10);
        $i = 1;
        foreach($kategori as $ktg) {
            $countitem = $this->db-
>get_where('tb_barang',['barang_kategori' =>
$ktg['kat_id']])>num_rows();
            $pdf->Cell(10,8,$i.',',1,0,'T');
            $pdf-
>Cell(96,8,$ktg['kat_nama'],1,0,'T');
            $pdf-
>Cell(90,8,$countitem,1,1,'T');
            $i++;
        }
        $pdf->Output('D',$title.'.pdf');
    }

    public function lap_barang() {
        $data = array (
            'title'
            => 'Laporan Data Barang',
            'me'
            => $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'barang'
            => $this->Owner_model-
>data_barang(),
        );
        $this->load-
>view('owner/header', $data);
        $this->load-
>view('owner/lap_barang', $data);
        $this->load-
>view('owner/footer');
    }

    public function lap_barang_print() {
        $data = array (
            'title'
            => 'Laporan Data Barang',
            'me'
            => $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'barang'
            => $this->Owner_model-

```

```

>data_barang(),
    );
    $this->load-
>view('owner/lap_barang_print', $data);
    }
    public function lap_stok_barang() {
        $data = array (
            'title'
=> 'Laporan Stok Barang',
            'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'barang'
=> $this->Owner_model-
>stok_barang(),
        );
        $this->load-
>view('owner/header', $data);
        $this->load-
>view('owner/lap_stok_barang', $data);
        $this->load-
>view('owner/footer');
    }
    public function lap_stok_barang_print()
    {
        $data = array (
            'title'
=> 'Laporan Stok Barang',
            'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'barang'
=> $this->Owner_model-
>stok_barang(),
        );
        $this->load-
>view('owner/lap_stok_barang_print', $data);
    }
    public function
lap_stok_barang_excel() {
        $data = array (
            'title'
=> 'Laporan Stok Barang',
            'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'

```

```

=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'barang'
=> $this->Owner_model-
>stok_barang(),
        );
        $this->load-
>view('owner/lap_stok_barang_excel', $data);
    }

```

## 2. Admin.php

```

<?php
defined('BASEPATH') OR exit('No direct script
access allowed');

```

```

class Admin extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load-
>model('Admin_model');
        if($this->session-
>userdata('status') != 'admin_loggedin') {
            redirect("");
        }
    }
    public function dashboard() {
        $data = array (
            'title'
=> 'Dashboard',
            'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'kategori'
=> $this->db-
>get_where('tb_kategori',['kat_user' => $this-
>session->userdata('id')])>num_rows(),
            'barang'
=> $this->db-
>get_where('tb_barang',['barang_user' => $this-
>session->userdata('id')])>num_rows(),
            'supplier'
=> $this->db-
>get_where('tb_barang',['barang_user' => $this-
>session->userdata('id')])>num_rows(),
            'penjualan'
=> $this->db->get('tb_transaksi')-
>num_rows(),
            'grafikpenjualan'=>
        $this->Admin_model-
>grafik_penjualan(),
            'grafikkategori'=>

```

```

        $this->Admin_model-
>grafik_kategori(),
        );
        $this->load-
>view('admin/header', $data);
        $this->load-
>view('admin/index', $data);
        $this->load-
>view('admin/footer');
    }
    public function satuan() {
        $data = array (
            'title'
=>        'Satuan',
            'me'
=>        $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
=>        $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'satuan'
=>        $this->db->get('tb_satuan')-
>result_array(),
        );
        $this->load-
>view('admin/header', $data);
        $this->load-
>view('admin/satuan', $data);
        $this->load-
>view('admin/footer');
    }
    public function satuan_add() {
        $data = array (
            'title'
=>        'Input Satuan',
            'atr'
=>        $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'me'
=>        $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        );
        $this->form_validation-
>set_rules('nama', 'nama',
'required|is_unique[tb_satuan.satuan_nama]', [
            'required' =>        'Kolom ini tidak
boleh kosong',
            'is_unique' =>        'Nama
satuan sudah ada']);
        if($this->form_validation-
>run() == FALSE) {
            $this->load-

```

```

>view('admin/header', $data);
        $this->load-
>view('admin/satuan_add', $data);
        $this->load-
>view('admin/footer');
        }else {
            $this-
>Admin_model->simpan_satuan();
            $this->session-
>set_flashdata('flash', 'Data baru berhasil
ditambahkan');
            redirect('admin/master/satuan');
        }
    public function satuan_edit($id) {
        $data = array (
            'title'
=>        'Edit Satuan',
            'atr'
=>        $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'me'
=>        $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'satid'
=>        $this->db-
>get_where('tb_satuan',['satuan_id' => $id])-
>row_array(),
        );
        $this->form_validation-
>set_rules('nama', 'nama', 'required', [
            'required' =>        'Kolom ini tidak
boleh kosong']);
        if($this->form_validation-
>run() == FALSE) {
            $this->load-
>view('admin/header', $data);
            $this->load-
>view('admin/satuan_edit', $data);
            $this->load-
>view('admin/footer');
        }else {
            $this-
>Admin_model->edit_satuan($id);
            $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');
            redirect('admin/master/satuan');
        }
    public function satuan_del($id) {
        $this->db->where('satuan_id',

```

```

$id);
        $this->db->delete('tb_satuan');
        $this->session->destroy();
    }
    >set_flashdata('flash', 'Data berhasil dihapus');

    redirect('admin/master/satuan');
}
public function satuan_kosongkan() {
    $this->db->truncate('tb_satuan');
    $this->session->destroy();
    >set_flashdata('flash', 'Tabel berhasil dikosongkan');

    redirect('admin/master/satuan');
}
public function kategori() {
    $data = array (
        'title' => 'Kategori',
        'me' => $this->db->get_where('tb_users', ['user_id' => $this->session->userdata('id')])->row_array(),
        'atr' => $this->db->get_where('tb_atribut', ['id' => 1])->row_array(),
        'kategori' => $this->Admin_model->data_kategori(),
    );
    $this->load->view('admin/header', $data);
    $this->load->view('admin/kategori', $data);
    $this->load->view('admin/footer');
}
public function kategori_add() {
    $data = array (
        'title' => 'Input Kategori',
        'atr' => $this->db->get_where('tb_atribut', ['id' => 1])->row_array(),
        'me' => $this->db->get_where('tb_users', ['user_id' => $this->session->userdata('id')])->row_array(),
    );
    $this->form_validation->set_rules('nama', 'nama', 'required', [
        'required' => 'Kolom ini tidak boleh kosong']);
    if($this->form_validation->

```

```

run() == FALSE) {
    $this->load->view('admin/header', $data);
    $this->load->view('admin/kategori_add', $data);
    $this->load->view('admin/footer');
} else {
    $this->Admin_model->simpan_kategori();
    $this->session->destroy();
    >set_flashdata('flash', 'Data baru berhasil ditambahkan');

    redirect('admin/master/kategori');
}
public function kategori_edit($id) {
    $data = array (
        'title' => 'Edit Kategori',
        'atr' => $this->db->get_where('tb_atribut', ['id' => 1])->row_array(),
        'me' => $this->db->get_where('tb_users', ['user_id' => $this->session->userdata('id')])->row_array(),
        'katid' => $this->db->get_where('tb_kategori', ['kat_id' => $id])->row_array(),
    );
    $this->form_validation->set_rules('nama', 'nama', 'required', [
        'required' => 'Kolom ini tidak boleh kosong']);
    if($this->form_validation->run() == FALSE) {
        $this->load->view('admin/header', $data);
        $this->load->view('admin/kategori_edit', $data);
        $this->load->view('admin/footer');
    } else {
        $this->Admin_model->edit_kategori($id);
        $this->session->destroy();
        >set_flashdata('flash', 'Data berhasil diperbaharui');

        redirect('admin/master/kategori');
    }
}

```

```

        public function kategori_del($id) {
            $this->db->where('kat_id',
                $id);
            $this->db->delete('tb_kategori');
            $this->session->set_flashdata('flash', 'Data berhasil dihapus');
            redirect('admin/master/kategori');
        }
        public function kategori_kosongkan() {
            $this->db->where('kat_user',
                $this->session->userdata('id'));
            $this->db->delete('tb_kategori');
            $this->session->set_flashdata('flash', 'Tabel berhasil dikosongkan');
            redirect('admin/master/kategori');
        }
        public function barang() {
            $data = array (
                'title' => 'Barang',
                'me' => $this->db->get_where('tb_users', ['user_id' => $this->session->userdata('id')])->row_array(),
                'atr' => $this->db->get_where('tb_tribut', ['id' => 1])->row_array(),
                'barang' => $this->Admin_model->data_barang(),
            );
            $this->load->view('admin/header', $data);
            $this->load->view('admin/barang', $data);
            $this->load->view('admin/footer');
        }
        public function barang_add() {
            $data = array (
                'title' => 'Input Barang',
                'atr' => $this->db->get_where('tb_tribut', ['id' => 1])->row_array(),
                'me' => $this->db->get_where('tb_users', ['user_id' => $this->session->userdata('id')])->row_array(),
                'kategori' => $this->Admin_model->data_kategori(),
            );
            $this->load->view('admin/header', $data);
            $this->load->view('admin/barang_add', $data);
            $this->load->view('admin/footer');
        }
    }
}

```

```

        >data_kategori(),
            'supplier' => $this->Admin_model->data_supplier(),
            'satuan' => $this->db->get('tb_satuan')->result_array(),
        );
        $this->form_validation->set_rules('id', 'id', 'required|is_unique[tb_barang.barang_id]', [
            'required' => 'Kolom ini tidak boleh kosong',
            'is_unique' => 'Id barang sudah digunakan']);
        $this->form_validation->set_rules('nama', 'nama', 'required', [
            'required' => 'Kolom ini tidak boleh kosong']);
        $this->form_validation->set_rules('kategori', 'kategori', 'required', [
            'required' => 'Kolom ini tidak boleh kosong']);
        $this->form_validation->set_rules('harga', 'harga', 'required', [
            'required' => 'Kolom ini tidak boleh kosong']);
        $this->form_validation->set_rules('stok', 'stok', 'required', [
            'required' => 'Kolom ini tidak boleh kosong']);
        $this->form_validation->set_rules('supplier', 'supplier', 'required', [
            'required' => 'Kolom ini tidak boleh kosong']);
        $this->form_validation->set_rules('satuan', 'satuan', 'required', [
            'required' => 'Kolom ini tidak boleh kosong']);
        if($this->form_validation->run() == FALSE) {
            $this->load->view('admin/header', $data);
            $this->load->view('admin/barang_add', $data);
            $this->load->view('admin/footer');
        } else {

```

```

        $this->Admin_model->simpan_barang();
        $this->session->set_flashdata('flash', 'Data baru berhasil
ditambahkan');

        redirect('admin/master/barang');
    }
    public function barang_edit($id) {
        $data = array (
            'title'
=> 'Edit Barang',
            'atr'
=> $this->db->get_where('tb_attribut',['id' => 1])->row_array(),
            'me'
=> $this->db->get_where('tb_users',['user_id' => $this->session->userdata('id')])->row_array(),
            'kategori'
=> $this->Admin_model->data_kategori(),
            'supplier'
=> $this->Admin_model->data_supplier(),
            'barid'
=> $this->db->get_where('tb_barang',['barang_id' => $id])->row_array(),
            'satuan'
=> $this->db->get('tb_satuan')->result_array(),
        );
        $this->form_validation->set_rules('id', 'id', 'required', [
            'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation->set_rules('nama', 'nama', 'required', [
            'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation->set_rules('kategori', 'kategori', 'required', [
            'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation->set_rules('harga', 'harga', 'required', [
            'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation->set_rules('stok', 'stok', 'required', [

```

```

            'required'=> 'Kolom ini tidak
boleh kosong'];
        $this->form_validation->set_rules('satuan', 'satuan', 'required', [
            'required'=> 'Kolom ini tidak
boleh kosong'];
        if($this->form_validation->run() == FALSE) {
            $this->load->view('admin/header', $data);
            $this->load->view('admin/barang_edit', $data);
            $this->load->view('admin/footer');
        } else {
            $this->Admin_model->edit_barang($id);
            $this->session->set_flashdata('flash', 'Data berhasil
diperbaharui');

            redirect('admin/master/barang');
        }
        public function barang_del($id) {
            $this->db->where('barang_id',
$id);
            $this->db->delete('tb_barang');
            $this->session->set_flashdata('flash', 'Data berhasil
dihapus');

            redirect('admin/master/barang');
        }
        public function barang_kosongkan() {
            $this->db->where('barang_user', $this->session->userdata('id'));
            $this->db->delete('tb_barang');
            $this->session->set_flashdata('flash', 'Tabel berhasil
dikosongkan');

            redirect('admin/master/barang');
        }
        public function supplier() {
            $data = array (
                'title'
=> 'Supplier',
                'me'
=> $this->db->get_where('tb_users',['user_id' => $this->

```

```

>session->userdata('id'))->row_array(),
    'atr'
    => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
    'supplier'
    => $this->Admin_model-
>data_supplier(),
    );
    $this->load-
>view('admin/header', $data);
    $this->load-
>view('admin/supplier', $data);
    $this->load-
>view('admin/footer');
    }
    public function supplier_add() {
        $data = array (
            'title'
            => 'Input Supplier',
            'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'me'
            => $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
            );
            $this->form_validation-
>set_rules('nama', 'nama', 'required', [
                'required'=> 'Kolom ini tidak
                boleh kosong'];
                $this->form_validation-
>set_rules('username', 'username',
'required|is_unique[tb_supplier.supplier_usernam
e]', [
                'required'=> 'Kolom ini tidak
                boleh kosong',
                'is_unique' => 'Username
                ini sudah tersedia']);
                $this->form_validation-
>set_rules('alamat', 'alamat', 'required', [
                'required'=> 'Kolom ini tidak
                boleh kosong'];
                $this->form_validation-
>set_rules('nohp', 'nohp', 'required', [
                'required'=> 'Kolom ini tidak
                boleh kosong'];
                if($this->form_validation-
>run() == FALSE) {
                    $this->load-
>view('admin/header', $data);

```

```

    $this->load-
>view('admin/supplier_add', $data);
    $this->load-
>view('admin/footer');
    }else {
        $this-
>Admin_model->simpan_supplier();
        $this->session-
>set_flashdata('flash', 'Data baru berhasil
ditambahkan');
        redirect('admin/master/supplier');
    }
    public function supplier_edit($id) {
        $data = array (
            'title'
            => 'Edit Supplier',
            'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'me'
            => $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
            'supid'
            => $this->db-
>get_where('tb_supplier',['supplier_id' => $id])->
row_array(),
            );
            $this->form_validation-
>set_rules('nama', 'nama', 'required', [
                'required'=> 'Kolom ini tidak
                boleh kosong'];
                $this->form_validation-
>set_rules('username', 'username', 'required', [
                'required'=> 'Kolom ini tidak
                boleh kosong'];
                $this->form_validation-
>set_rules('alamat', 'alamat', 'required', [
                'required'=> 'Kolom ini tidak
                boleh kosong'];
                $this->form_validation-
>set_rules('nohp', 'nohp', 'required', [
                'required'=> 'Kolom ini tidak
                boleh kosong'];
                if($this->form_validation-
>run() == FALSE) {
                    $this->load-
>view('admin/header', $data);
                    $this->load-
>view('admin/supplier_edit', $data);

```

```

        $this->load-
>view('admin/footer');
        }else {
            $this-
>Admin_model->edit_supplier($id);
            $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

            redirect('admin/master/supplier');
        }
        public function supplier_del($id) {
            $this->db-
>where('supplier_id', $id);
            $this->db-
>delete('tb_supplier');
            $this->session-
>set_flashdata('flash', 'Data berhasil dihapus');

            redirect('admin/master/supplier');
        }
        public function supplier_kosongkan() {
            $this->db-
>truncate('tb_supplier');
            $this->session-
>set_flashdata('flash', 'Tabel berhasil
dikosongkan');

            redirect('admin/master/supplier');
        }
        public function akses() {
            $data = array (
                'title'
=> 'Akses',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')]->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'akses'
=> $this->Admin_model-
>data_akses(),
            );
            $this->load-
>view('admin/header', $data);
            $this->load-
>view('admin/akses', $data);
            $this->load-
>view('admin/footer');
        }
        public function akses_add() {
            $data = array (
                'title'
=> 'Input Akses',
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
            );
            $this->form_validation-
>set_rules('nama', 'nama', 'required', [
                'required'=> 'Kolom ini tidak
boleh kosong'];
            $this->form_validation-
>set_rules('username', 'username',
'required|is_unique[tb_users.user_username]', [
                'required'=> 'Kolom ini tidak
boleh kosong',
                'is_unique' => 'Username
ini sudah tersedia'];
            $this->form_validation-
>set_rules('level', 'level', 'required', [
                'required'=> 'Kolom ini tidak
boleh kosong'];
            if($this->form_validation-
>run() == FALSE) {
                $this->load-
>view('admin/header', $data);
                $this->load-
>view('admin/akses_add', $data);
                $this->load-
>view('admin/footer');
            }else {
                $this-
>Admin_model->simpan_akses();
                $this->session-
>set_flashdata('flash', 'Data baru berhasil
ditambahkan');

                redirect('admin/akses');
            }
        }
        public function akses_edit($id) {
            $data = array (
                'title'
=> 'Edit Akses',
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'me'
=> $this->db-

```

```

>get_where('tb_users',['user_id' => $this-
>session->userdata('id')]->row_array(),
        'aksid'
        => $this->db-
>get_where('tb_users',['user_id' => $id])-
>row_array(),
        );
        $this->form_validation-
>set_rules('nama', 'nama', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation-
>set_rules('username', 'username', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation-
>set_rules('level', 'level', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong']);
        if($this->form_validation-
>run() == FALSE) {
                $this->load-
>view('admin/header', $data);
                $this->load-
>view('admin/akses_edit', $data);
                $this->load-
>view('admin/footer');
        } else {
                $this-
>Admin_model->edit_akses($id);
                $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

                redirect('admin/akses');
        }

        public function akses_del($id) {
                $this->db->where('user_id',
$id);
                $this->db->delete('tb_users');
                $this->session-
>set_flashdata('flash', 'Data berhasil dihapus');
                redirect('admin/akses');
        }

        public function penjualan() {
                $data = array (
                        'title'
=> 'Penjualan',
                        'me'
=> $this->db-

```

```

>get_where('tb_users',['user_id' => $this-
>session->userdata('id')]->row_array(),
        'atr'
        => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'transaksi'
        => $this->Admin_model-
>data_transaksi(),
        );
        $this->load-
>view('admin/header', $data);
        $this->load-
>view('admin/transaksi', $data);
        $this->load-
>view('admin/footer');
    }

    public function penjualan_add() {
        $data = array (
                'title'
=> 'Transaksi Baru',
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')]->row_array(),
                'barang'
=> $this->Admin_model-
>data_barang(),
                'keranjang'
=> $this->cart->contents(),
        );
        $this->form_validation-
>set_rules('kode', 'kode', 'required', [
                'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation-
>set_rules('jml', 'jml', 'required', [
                'required'=> 'Kolom ini tidak
boleh kosong']);
        if($this->form_validation-
>run() == FALSE) {
                $this->load-
>view('admin/header', $data);
                $this->load-
>view('admin/transaksi_add', $data);
                $this->load-
>view('admin/footer');
        } else {
                $cek = $this->db-
>get_where('tb_barang',['barang_id' => $this-
>input->post('kode')])->row_array();

```

```

        if($this->input-
>post('jml') > $cek['barang_stok']) {
            $this-
>session->set_flashdata('flashnotif', 'Stok
'. $cek['barang_nama']. ' sisa
'. $cek['barang_stok']);

            redirect('admin/penjualan/new');
        }
        $dcart = array (
            'id'
=> $cek['barang_id'],
            'name'
=> $cek['barang_nama'],
            'price'
=> $cek['barang_harga'],
            'qty'
=> $this->input->post('jml'),
        );
        $this->cart-
>insert($dcart);
        $this->session-
>set_flashdata('flash', 'Barang berhasil
ditambahkan dalam keranjang');

        redirect('admin/penjualan/new');
    }

    public function cart_trash($id) {
        $this->cart->remove($id);

        redirect('admin/penjualan/new');
    }

    public function cart_kosong() {
        $this->cart->destroy();

        redirect('admin/penjualan/new');
    }

    public function simpan_transaksi() {
        $this->db-
>order_by('transaksi_urut', 'DESC');
        $sqlinc = $this->db-
>get_where('tb_transaksi', ['transaksi_tgl' =>
date('Y-m-d')]);
        if ($sqlinc->num_rows() ==
0) {
            $autoinc = 1;
        } else {
            $urutinc = $sqlinc-
>row()->transaksi_urut;
            $urutinc++;
            $autoinc = $urutinc;
        }
    }

```

```

        $this->db-
>order_by('transaksi_urut', 'DESC');
        $sql = $this->db-
>get_where('tb_transaksi', ['transaksi_tgl' =>
date('Y-m-d')]);
        if ($sql->num_rows() == 0) {
            $trax_kode =
date('ymd')."0001";
        } else {
            $noUrut = substr($sql-
>row()->transaksi_kode, 6, 4);
            $noUrut++;
            $trax_kode =
date('ymd').sprintf("%04s", $noUrut);
        }
        $traxid = md5(rand());
        $bayar = preg_replace('/[^0-
9]/', "", $this->input->post('bayar'));
        $kembali =
preg_replace('/[^0-9]/', "", $this->input-
>post('kembalian'));
        $data = array (
            'transaksi_id'
=> $traxid,
            'transaksi_kode'
=> $trax_kode,
            'transaksi_urut'
=> $autoinc,
            'transaksi_tgl'
=> date('Y-m-d'),
            'transaksi_total'
=> $this->cart->total(),
            'transaksi_bayar'
=> $bayar,
            'transaksi_kembalian' =>
$kembali,
        );
        if($this->db-
>insert('tb_transaksi', $data)) {
            foreach($this->cart-
>contents() as $cart) {
                $detail =
array (
                    'dtrax_id'
=>
                    $traxid,
                    'dtrax_item'
=>
                    $cart['id'],
                    'dtrax_harga'
=>
                    $cart['price'],
                    'dtrax_qty'

```

```

=> $cart['qty'],
'dtrax_subtotal' =>
$cart['subtotal'],
);
$this->db-
>insert('tb_detail_transaksi', $detail);
}
}
$this->cart->destroy();
$this->session-
>set_flashdata('flash', 'Transaksi berhasil');
redirect('admin/penjualan');
}

public function penjualan_del($id) {
$this->db-
>where('transaksi_id', $id);
$this->db-
>delete('tb_transaksi');
$this->session-
>set_flashdata('flash', 'Data berhasil dihapus');
redirect('admin/penjualan');
}

public function stok_barang() {
$data = array (
'title'
=> 'Stok Barang',
'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
'barang'
=> $this->Admin_model-
>stok_barang(),
);
$this->load-
>view('admin/header', $data);
$this->load-
>view('admin/stok_barang', $data);
$this->load-
>view('admin/footer');
}

public function
stok_barang_request($id) {
$data = array (
'title'
=> 'Input Request',
'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),

```

```

'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
'barid'
=> $this->db-
>get_where('tb_barang',['barang_id' => $id])-
>row_array(),
);
$this->form_validation-
>set_rules('stok', 'stok', 'required', [
'required'=> 'Kolom ini tidak
boleh kosong']);
if($this->form_validation-
>run() == FALSE) {
$this->load-
>view('admin/header', $data);
$this->load-
>view('admin/stok_barang_add', $data);
$this->load-
>view('admin/footer');
}else {
$this-
>Admin_model->simpan_request($id);
// $this->session-
>set_flashdata('flash', 'Permintaan stok berhasil
dikirim');
//
redirect('admin/daftar-pemesanan');
}

public function list_pemesanan() {
$data = array (
'title'
=> 'Daftar Pemesanan',
'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
'pemesanan'
=> $this->Admin_model-
>data_pemesanan(),
);
$this->load-
>view('admin/header', $data);
$this->load-
>view('admin/pemesanan', $data);
$this->load-
>view('admin/footer');
}

```

```

        public function
list_pemesanan_kosongkan() {
    $this->db-
>where('pemesn_user', $this->session-
>userdata('id'));
    $this->db-
>delete('tb_pemesanan');
    $this->session-
>set_flashdata('flash', 'Tabel berhasil
dikosongkan');
    redirect('admin/daftar-
pemesanan');
}

        public function
list_pemesanan_batal($id) {
    $cek = $this->db-
>get_where('tb_pemesanan',['pemesn_id' =>
$id])->row_array();
    $this->db-
>set('pemesn_status', 'Dibatalkan');
    $this->db-
>where('pemesn_id', $id);
    if($this->db-
>update('tb_pemesanan')) {
        $notif = array (
            'notif_id'
=> md5(rand()),
            'notif_dari'
=> $this->session->userdata('id'),
            'notif_ke'
=> $cek['pemesn_supplier'],
            'notif_isi'
=> 'Permintaan stok barang
dibatalkan',
            'notif_status' => 'Belum',
            'notif_pemesn' => ",
        );
        $this->db-
>insert('tb_notifikasi', $notif);
    }
    $this->session-
>set_flashdata('flash', 'Permintaan stok
dibatalkan');
    redirect('admin/daftar-
pemesanan');
}

        public function
list_pemesanan_terima($id) {
    $cek = $this->db-
>get_where('tb_pemesanan',['pemesn_id' =>
$id])->row_array();
    $cekbrg = $this->db-

```

```

>get_where('tb_barang',['barang_id_supp' =>
$cek['pemesn_barang']])->row_array();
    $stokbrg =
$cekbrg['barang_stok']+ $cek['pemesn_stok'];
    $this->db-
>set('pemesn_status', 'Diterima');
    $this->db-
>where('pemesn_id', $id);
    if($this->db-
>update('tb_pemesanan')) {
        $this->db-
>set('barang_stok', $stokbrg);
        $this->db-
>where('barang_id_supp',
$cek['pemesn_barang']);
        $this->db-
>update('tb_barang');
    }
    $this->session-
>set_flashdata('flash', 'Barang diterima');
    redirect('admin/daftar-
pemesanan');
}

        public function uploadbukti() {
            // get foto
            $config['upload_path'] =
'./assets/bukti-bayar/';
            $config['allowed_types'] =
'jpg|png|jpeg|gif';
            $config['encrypt_name'] = TRUE;

            $this->upload->initialize($config);
            if
(!empty($_FILES['gambar']['name'])) {
                if ( $this->upload-
>do_upload('gambar') ) {
                    $gambar = $this->upload-
>data();

                    $data = array(
                        'pemesn_bukti_bayar' =>
$gambar['file_name']
                    );
                    $this->db-
>where('pemesn_id', $this->input->post('id'));
                    $this->db-
>update('tb_pemesanan', $data);
                    $this->session-
>set_flashdata('flash', 'Bukti pembayaran berhasil
diupload');

                    redirect('admin/daftar-pemesanan');
                }
            }

```

```

        }else {
            $this->session->
>set_flashdata('error', 'Gagal upload bukti
pembayaran, kolom foto tidak boleh kosong. ');

            redirect('admin/daftar-pemesanan');
        }
    }

    public function list_penawaran() {
        $data = array (
            'title'
=> 'Penawaran Barang',
            'me'
=> $this->db->
>get_where('tb_users',['user_id' => $this->
>session->userdata('id')])>row_array(),
            'atr'
=> $this->db->
>get_where('tb_atribut',['id' => 1])>row_array(),
            'barang'
=> $this->Admin_model->
>data_penawaran(),
        );
        $this->load->
>view('admin/header', $data);
        $this->load->
>view('admin/penawaran', $data);
        $this->load->
>view('admin/footer');
    }

    public function
list_penawaran_request($id) {
        $data = array (
            'title'
=> 'Input Request Barang',
            'atr'
=> $this->db->
>get_where('tb_atribut',['id' => 1])>row_array(),
            'me'
=> $this->db->
>get_where('tb_users',['user_id' => $this->
>session->userdata('id')])>row_array(),
            'barid'
=> $this->db->
>get_where('tb_penawaran',['penawaran_id' =>
$id])>row_array(),
        );
        $this->form_validation->
>set_rules('stok', 'stok', 'required', [
            'required'=> 'Kolom ini tidak
boleh kosong']);
        if($this->form_validation->

```

```

>run() == FALSE) {
            $this->load->
>view('admin/header', $data);
            $this->load->
>view('admin/penawaran_add', $data);
            $this->load->
>view('admin/footer');
        }else {
            $this->
>Admin_model->request_barang($id);
            // $this->session->
>set_flashdata('flash', 'Permintaan barang baru
berhasil dikirim');
            //
            redirect('admin/daftar-pemesanan');
        }
    }

    public function
list_penawaran_tolak($id) {
        $data = array (
            'penact_penawaran'
=> $id,
            'penact_toko'
=> $this->session->userdata('id'),
            'penact_status'
=> 1,
        );
        $this->db->
>insert('tb_penawaran_action', $data);
        redirect('admin/penawaran-
barang');
    }

    public function list_pesanan() {
        $data = array (
            'title'
=> 'Data Pesanan Barang',
            'me'
=> $this->db->
>get_where('tb_users',['user_id' => $this->
>session->userdata('id')])>row_array(),
            'atr'
=> $this->db->
>get_where('tb_atribut',['id' => 1])>row_array(),
            'barang'
=> $this->Admin_model->
>data_pesanan(),
        );
        $this->load->
>view('admin/header', $data);
        $this->load->
>view('admin/pesanan_barang', $data);
        $this->load->
>view('admin/footer');
    }

```

```

    }

    public function list_pesanan_add() {
        $data = array (
            'title'
=>      'Input Pesanan Barang',
            'atr'
=>      $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'me'
=>      $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
            'supplier'
=>      $this->Admin_model-
>data_supplier(),
            'satuan'
=>      $this->db->get('tb_satuan')-
>result_array(),
        );
        $this->form_validation-
>set_rules('nama', 'nama',
'required|is_unique[tb_pesanan_barang.pesbar_barang]', [
            'required'=>      'Kolom ini tidak
boleh kosong',
            'is_unique' =>      'Pesanan
untuk barang ini sudah pernah');
        $this->form_validation-
>set_rules('stok', 'stok', 'required', [
            'required'=>      'Kolom ini tidak
boleh kosong'];
        $this->form_validation-
>set_rules('supplier', 'supplier', 'required', [
            'required'=>      'Kolom ini tidak
boleh kosong'];
        $this->form_validation-
>set_rules('satuan', 'satuan', 'required', [
            'required'=>      'Kolom ini tidak
boleh kosong'];
        if($this->form_validation-
>run() == FALSE) {
            $this->load-
>view('admin/header', $data);
            $this->load-
>view('admin/pesanan_add', $data);
            $this->load-
>view('admin/footer');
        }else {
            $this-
>Admin_model->simpan_pesanan();

```

```

        $this->session-
>set_flashdata('flash', 'Data baru berhasil
ditambahkan');
        redirect('admin/pesanan-barang');
    }

    public function list_pesanan_edit($id) {
        $data = array (
            'title'
=>      'Edit Pesanan Barang',
            'atr'
=>      $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'me'
=>      $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
            'supplier'
=>      $this->Admin_model-
>data_supplier(),
            'satuan'
=>      $this->db->get('tb_satuan')-
>result_array(),
            'pesid'
=>      $this->db-
>get_where('tb_pesanan_barang',['pesbar_id' =>
$id])->row_array(),
        );
        $this->form_validation-
>set_rules('nama', 'nama', 'required', [
            'required'=>      'Kolom ini tidak
boleh kosong'];
        $this->form_validation-
>set_rules('stok', 'stok', 'required', [
            'required'=>      'Kolom ini tidak
boleh kosong'];
        $this->form_validation-
>set_rules('supplier', 'supplier', 'required', [
            'required'=>      'Kolom ini tidak
boleh kosong'];
        $this->form_validation-
>set_rules('satuan', 'satuan', 'required', [
            'required'=>      'Kolom ini tidak
boleh kosong'];
        if($this->form_validation-
>run() == FALSE) {
            $this->load-
>view('admin/header', $data);
            $this->load-
>view('admin/pesanan_edit', $data);

```

```

        $this->load-
>view('admin/footer');
        }else {
            $this-
>Admin_model->edit_pesanan($id);
            $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

```

```

        redirect('admin/pesan-barang');
    }
}

public function list_pesanan_del($id) {
    $this->db->where('pesbar_id',
$id);
    $this->db-
>delete('tb_pesanan_barang');
    $this->session-
>set_flashdata('flash', 'Data berhasil
dihapus');
    redirect('admin/pesan-
barang');
}

public function
list_pesanan_kosongkan() {
    $this->db-
>truncate('tb_pesanan_barang');
    $this->session-
>set_flashdata('flash', 'Tabel berhasil
dikosongkan');
    redirect('admin/pesan-
barang');
}

public function uploadbuktipesbar() {

    // get foto
    $config['upload_path'] =
'./assets/bukti-bayar/';
    $config['allowed_types'] =
'jpg|png|jpeg|gif';
    $config['encrypt_name'] = TRUE;

    $this->upload->initialize($config);
    if
(!empty($_FILES['gambar']['name'])) {
        if ( $this->upload-
>do_upload('gambar') ) {
            $gambar = $this->upload-
>data();

            $data = array(

                'pesbar_bukti_bayar' =>
                $gambar['file_name']

```

```

        );
        $this->db->where('pesbar_id',
$this->input->post('id'));
        $this->db-
>update('tb_pesanan_barang', $data);
        $this->session-
>set_flashdata('flash', 'Bukti pembayaran berhasil
diupload');

```

```

        redirect('admin/pesan-barang');
    }
}else {
    $this->session-
>set_flashdata('error', 'Gagal upload bukti
pembayaran, kolom foto tidak boleh kosong. ');

    redirect('admin/pesan-barang');
}

public function pesbar_terima($id) {
    $this->db-
>update('tb_pesanan_barang',['pesbar_status' =>
'Diterima'],array('pesbar_id' => $id));
    $this->session-
>set_flashdata('flash', 'Barang diterima');
    redirect('admin/pesan-
barang');
}

public function lap_kategori() {
    $data = array (
        'title'
=> 'Laporan Data Kategori',
        'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
        'kategori'
=> $this->Admin_model-
>data_kategori(),
    );
    $this->load-
>view('admin/header', $data);
    $this->load-
>view('admin/lap_kategori', $data);
    $this->load-
>view('admin/footer');
}

public function lap_kategori_print() {
    $data = array (

```

```

                'title'
=> 'Laporan Data Kategori',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'kategori'
=> $this->Admin_model-
>data_kategori(),
                );
                $this->load-
>view('admin/lap_kategori_print', $data);
        }

        public function lap_kategori_excel() {
            $data = array (
                'title'
=> 'Laporan Data Kategori',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'kategori'
=> $this->Admin_model-
>data_kategori(),
                );
                $this->load-
>view('admin/lap_kategori_excel', $data);
        }

        public function lap_kategori_pdf() {
            $title = 'Laporan Data
Kategori';
            $atr = $this->db-
>get_where('tb_atribut',['id' => 1])->row_array();
            $kategori = $this-
>Admin_model->data_kategori();
            $pdf = new
FPDF('p','mm','Legal');
            // membuat halaman baru
            $pdf->AddPage();
            // setting jenis font yang akan digunakan
            $pdf->SetFillColor(112, 226, 137);
            $pdf-
>Image(base_url()).'assets2/images/'. $atr['favicon'
], 30, 12, -300);
            $pdf->SetFont('Arial','B',16);
            // mencetak string
            $pdf->Cell(195,7,$title,0,1,'C');
            $pdf->SetFont('Arial','B',14);

```

```

            $pdf->Cell(195,7,$atr['nama'],0,1,'C');
            $pdf->SetFont('Arial','B',12);
            $pdf->Cell(195,7,$atr['alamat'],0,1,'C');
            $pdf->Ln(6);
            $pdf->SetFont('Times','B',11);
            $pdf->Cell(10,8,'No',1,0,'C',1);
            $pdf->Cell(96,8,'Kategori',1,0,'C',1);
            $pdf->Cell(90,8,'Jumlah Barang',1,1,'C',1);

            $pdf->SetFont('Times','',10);
            $i = 1;
            foreach($kategori as $ktg) {
                $countitem = $this->db-
>get_where('tb_barang',['barang_kategori' =>
$ktg['kat_id']])->num_rows();
                $pdf->Cell(10,8,$i.'.',1,0,'T');
                $pdf-
>Cell(96,8,$ktg['kat_nama'],1,0,'T');
                $pdf-
>Cell(90,8,$countitem,1,1,'T');
                $i++;
            }
            $pdf->Output('D',$title.'.pdf');
        }

        public function lap_barang() {
            $data = array (
                'title'
=> 'Laporan Data Barang',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'barang'
=> $this->Admin_model-
>data_barang(),
                );
                $this->load-
>view('admin/header', $data);
                $this->load-
>view('admin/lap_barang', $data);
                $this->load-
>view('admin/footer');
        }

        public function lap_barang_print() {
            $data = array (
                'title'
=> 'Laporan Data Barang',
                'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),

```

```

        'atr'
        => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'barang'
        => $this->Admin_model-
>data_barang(),
        );
        $this->load-
>view('admin/lap_barang_print', $data);
    }

    public function lap_barang_excel() {
        $data = array (
            'title'
            => 'Laporan Data Barang',
            'me'
            => $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'barang'
            => $this->Admin_model-
>data_barang(),
        );
        $this->load-
>view('admin/lap_barang_excel', $data);
    }

    public function lap_barang_pdf() {
        $title = 'Laporan Data
Barang';
        $atr = $this->db-
>get_where('tb_atribut',['id' => 1])->row_array();
        $barang = $this-
>Admin_model->data_barang();
        $pdf = new
FPDF('I','mm','Legal');
        // membuat halaman baru
        $pdf->AddPage();
        // setting jenis font yang akan digunakan
        $pdf->SetFont('Arial','B',16);
        $pdf-
>Image(base_url(). 'assets2/images/'. $atr['favicon'
], 50, 12, -300);
        $pdf->SetFont('Arial','B',16);
        // mencetak string
        $pdf->Cell(330,7,$title,0,1,'C');
        $pdf->SetFont('Arial','B',14);
        $pdf-
>Cell(330,7,$atr['nama'],0,1,'C');
        $pdf->SetFont('Arial','B',12);
        $pdf-
>Cell(330,7,$atr['alamat'],0,1,'C');
        $pdf->Ln(6);

```

```

        $pdf->SetFont('Times','B',11);
        $pdf->Cell(15,8,'No',1,0,'C',1);
        $pdf->Cell(100,8,'Nama',1,0,'C',1);
        $pdf->Cell(55,8,'Kategori',1,0,'C',1);
        $pdf->Cell(60,8,'Harga',1,0,'C',1);
        $pdf->Cell(35,8,'Stok',1,0,'C',1);
        $pdf->Cell(70,8,'Supplier',1,1,'C',1);

        $pdf->SetFont('Times','',10);
        $i = 1;
        foreach($barang as $brg) {
            $pdf->Cell(15,8,$i.',',1,0,'T');
            $pdf-
>Cell(100,8,$brg['barang_nama'],1,0,'T');
            $pdf-
>Cell(55,8,$brg['kat_nama'],1,0,'T');
            $pdf-
>Cell(60,8,number_format($brg['barang_harga'],
0,',','.'),1,0,'T');
            $pdf-
>Cell(35,8,number_format($brg['barang_stok'],0,
',','.'). $brg['satuan_nama'],1,0,'T');
            $pdf-
>Cell(70,8,$brg['supplier_nama'],1,1,'T');
            $i++;
        }
        $pdf->Output('D',$title.'.pdf');
    }

    public function lap_stok_barang() {
        $data = array (
            'title'
            => 'Laporan Stok Barang',
            'me'
            => $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'barang'
            => $this->Admin_model-
>stok_barang(),
        );
        $this->load-
>view('admin/header', $data);
        $this->load-
>view('admin/lap_stok_barang', $data);
        $this->load-
>view('admin/footer');
    }

    public function lap_stok_barang_print()
    {
        $data = array (
            'title'

```

```

=> 'Laporan Stok Barang',
    'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>->row_array(),
    'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>->row_array(),
    'barang'
=> $this->Admin_model-
>stok_barang(),
    );
    $this->load-
>view('admin/lap_stok_barang_print', $data);
    }

    public function
lap_stok_barang_excel() {
        $data = array (
            'title'
=> 'Laporan Stok Barang',
            'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>->row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>->row_array(),
            'barang'
=> $this->Admin_model-
>stok_barang(),
            );
            $this->load-
>view('admin/lap_stok_barang_excel', $data);
        }

    public function lap_stok_barang_pdf()
    {
        $title = 'Laporan Stok Barang';
        $atr = $this->db-
>get_where('tb_atribut',['id' => 1])>->row_array();
        $barang = $this-
>Admin_model->stok_barang();
        $pdf = new
FPDF('p','mm','Legal');
        // membuat halaman baru
        $pdf->AddPage();
        // setting jenis font yang akan digunakan
        $pdf->SetFont('Arial','B',16);
        // mencetak string
        $pdf->Cell(195,7,$title,0,1,'C');
        $pdf->SetFont('Arial','B',14);

```

```

        $pdf-
>Cell(195,7,$atr['nama'],0,1,'C');
        $pdf->SetFont('Arial','B',12);
        $pdf-
>Cell(195,7,$atr['alamat'],0,1,'C');
        $pdf->Ln(6);
        $pdf->SetFont('Times','B',11);
        $pdf->Cell(10,8,'No',1,0,'C',1);
        $pdf->Cell(116,8,'Nama Barang',1,0,'C',1);
        $pdf->Cell(70,8,'Stok',1,1,'C',1);

        $pdf->SetFont('Times','',10);
        $i = 1;
        foreach($barang as $brg) {
            if($brg['barang_stok'] <
$atr['batas'] OR $brg['barang_stok'] ==
$atr['batas']) {
                $pdf-
>SetTextColor(255, 0, 0);
                $pdf-
>Cell(10,8,$i.'.',1,0,'T');
                $pdf-
>Cell(116,8,$brg['barang_nama'],1,0,'T');
                $pdf-
>Cell(70,8,number_format($brg['barang_stok'],0,
';','').'.$brg['satuan_nama'],1,1,'T');
            }else {
                $pdf-
>SetTextColor(0, 0, 0);
                $pdf-
>Cell(10,8,$i.'.',1,0,'T');
                $pdf-
>Cell(116,8,$brg['barang_nama'],1,0,'T');
                $pdf-
>Cell(70,8,number_format($brg['barang_stok'],0,
';','').'.$brg['satuan_nama'],1,1,'T');
            }
            $i++;
        }
        $pdf->Output('D',$title.'.pdf');
    }

    public function lap_penjualan() {
        $start = $this->input-
>post('start');
        $end = $this->input-
>post('end');
        if($start == " AND $end == ")
        {
            $transaksi = $this-
>Admin_model->data_transaksi();
        }else {
            $transaksi = $this-
>Admin_model-
>data_transaksi_sortir(array($start,$end));
        }
    }

```

```

                $data = array (
                    'title'
=>        'Laporan Data Penjualan',
                    'me'
=>        $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')]->row_array(),
                    'atr'
=>        $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                    'transaksi'
=>        $transaksi,
                    'start'
=>        $start,
                    'end'
=>        $end,
                );
                $this->load-
>view('admin/header', $data);
                $this->load-
>view('admin/lap_transaksi', $data);
                $this->load-
>view('admin/footer');
            }

            public function lap_penjualan_print() {
                $start = $this->uri-
>segment(5);
                $end = $this->uri-
>segment(6);
                if($start == " AND $end == ")
            {
                    $transaksi = $this-
>Admin_model->data_transaksi();
                }else {
                    $transaksi = $this-
>Admin_model-
>data_transaksi_sortir(array($start,$end));
                }
                $data = array (
                    'title'
=>        'Laporan Data Penjualan',
                    'me'
=>        $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')]->row_array(),
                    'atr'
=>        $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                    'transaksi'
=>        $transaksi,
                    'start'
=>        $start,
                    'end'
=>        $end,
                );
            }

```

```

                $this->load-
>view('admin/lap_transaksi_print', $data);
            }

            public function lap_penjualan_excel() {
                $start = $this->uri-
>segment(5);
                $end = $this->uri-
>segment(6);
                if($start == " AND $end == ")
            {
                    $transaksi = $this-
>Admin_model->data_transaksi();
                }else {
                    $transaksi = $this-
>Admin_model-
>data_transaksi_sortir(array($start,$end));
                }
                $data = array (
                    'title'
=>        'Laporan Data Penjualan',
                    'me'
=>        $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')]->row_array(),
                    'atr'
=>        $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                    'transaksi'
=>        $transaksi,
                    'start'
=>        $start,
                    'end'
=>        $end,
                );
                $this->load-
>view('admin/lap_transaksi_excel', $data);
            }

            public function lap_penjualan_pdf() {
                $start = $this->uri-
>segment(5);
                $end = $this->uri-
>segment(6);
                if($start == " AND $end == ")
            {
                    $transaksi = $this-
>Admin_model->data_transaksi();
                }else {
                    $transaksi = $this-
>Admin_model-
>data_transaksi_sortir(array($start,$end));
                }
                $title = 'Laporan Data
Penjualan';
                $atr = $this->db-

```

```

>get_where('tb_atribut',['id' => 1])->row_array();
    $pdf = new
FPDF('p','mm','Legal');
    // membuat halaman baru
    $pdf->AddPage();
    // setting jenis font yang akan digunakan
    $pdf->SetFillColor(112, 226, 137);
    $pdf-
>Image(base_url()).'assets2/images/'. $atr['favicon'
], 30, 12, -300);
    $pdf->SetFont('Arial','B',16);
    // mencetak string
    if($start == " AND $end == ") {
    $pdf->Cell(195,7,$title,0,1,'C');
    $pdf->SetFont('Arial','B',14);
    $pdf-
>Cell(195,7,$atr['nama'],0,1,'C');
    $pdf->SetFont('Arial','B',12);
    $pdf-
>Cell(195,7,$atr['alamat'],0,1,'C');
    }else {
    $pdf->Cell(195,7,$title,0,1,'C');
    $pdf->SetFont('Arial','B',14);
    $pdf->Cell(195,7,'Periode '.date('d-m-
Y', strtotime($start)).' S/D '.date('d-m-Y',
strtotime($end)),0,1,'C');
    $pdf->SetFont('Arial','B',14);
    $pdf-
>Cell(195,7,$atr['nama'],0,1,'C');
    $pdf->SetFont('Arial','B',12);
    $pdf-
>Cell(195,7,$atr['alamat'],0,1,'C');
    }
    $pdf->Ln(6);
    $pdf->SetFont("Times','B',11);
    $pdf->Cell(10,8,'No',1,0,'C',1);
    $pdf->Cell(58,8,'Tanggal',1,0,'C',1);
    $pdf->Cell(78,8,'Kode Transaksi',1,0,'C',1);
    $pdf->Cell(50,8,'Total',1,1,'C',1);

    $pdf->SetFont("Times",",10);
    $i = 1;
    $total = 0;
    foreach($transaksi as $strax) {
        $total +=
    $strax['transaksi_total'];
        $pdf->Cell(10,8,$i.'!',1,0,'T');
        $pdf->Cell(58,8,date('d-m-Y',
strtotime($strax['transaksi_tgl'])),1,0,'T');
        $pdf-
>Cell(78,8,$strax['transaksi_kode'],1,0,'T');
        $pdf-
>Cell(50,8,number_format($strax['transaksi_total'
],0,',','.'),1,1,'T');
        $i++;
    }

```

```

    $pdf->SetFont('Times','B',11);
    $pdf->Cell(146,8,'Total',1,0,'C',1);
    $pdf-
>Cell(50,8,number_format($total,0,',','.'),1,1,'C',1
);
    $pdf->Output('D',$title.'.pdf');
    }

    public function lap_pemesanan() {
        $start = $this->input-
>post('start');
        $end = $this->input-
>post('end');
        if($start == " AND $end == ")
        {
            $pemesanan = $this-
>Admin_model->data_pemesanan();
        }else {
            $pemesanan = $this-
>Admin_model-
>data_pemesanan_sortir(array($start,$end));
        }
        $data = array (
            'title'
=> 'Laporan Data Pemesanan',
            'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'pemesanan'
=> $pemesanan,
            'start'
=> $start,
            'end'
=> $end,
        );
        $this->load-
>view('admin/header', $data);
        $this->load-
>view('admin/lap_pemesanan', $data);
        $this->load-
>view('admin/footer');
    }

    public function lap_pemesanan_print()
    {
        $start = $this->uri-
>segment(5);
        $end = $this->uri-
>segment(6);
        if($start == " AND $end == ")
        {
            $pemesanan = $this-

```

```

>Admin_model->data_pemesanan();
    }else {
        $pemesanan = $this-
>Admin_model-
>data_pemesanan_sortir(array($start,$end));
    }
    $data = array (
        'title'
=> 'Laporan Data Pemesanan',
        'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
        'pemesanan'
=> $pemesanan,
        'start'
=> $start,
        'end'
=> $end,
    );
    $this->load-
>view('admin/lap_pemesanan_print', $data);
    }

    public function lap_pemesanan_excel()
    {
        $start = $this->uri-
>segment(5);
        $end = $this->uri-
>segment(6);
        if($start == " AND $end == ")
        {
            $pemesanan = $this-
>Admin_model->data_pemesanan();
        }else {
            $pemesanan = $this-
>Admin_model-
>data_pemesanan_sortir(array($start,$end));
        }
        $data = array (
            'title'
=> 'Laporan Data Pemesanan',
            'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'pemesanan'
=> $pemesanan,
            'start'
=> $start,
            'end'
=> $end,
        );
        $this->load-
>view('admin/lap_pemesanan_excel', $data);
    }

    public function lap_pemesanan_pdf() {
        $start = $this->uri-
>segment(5);
        $end = $this->uri-
>segment(6);
        if($start == " AND $end == ")
        {
            $pemesanan = $this-
>Admin_model->data_pemesanan();
        }else {
            $pemesanan = $this-
>Admin_model-
>data_pemesanan_sortir(array($start,$end));
        }
        $title = 'Laporan Data
Pemesanan';
        $atr = $this->db-
>get_where('tb_atribut',['id' => 1])>row_array();
        $pdf = new
FPDF('l','mm','Legal');
        // membuat halaman baru
        $pdf->AddPage();
        // setting jenis font yang akan digunakan
        $pdf->SetFont('Arial', 'B', 12);
        $pdf->SetFillColor(112, 226, 137);
        $pdf->Image(base_url(). 'assets2/images/'. $atr['favicon']
, 50, 12, -300);
        $pdf->SetFont('Arial', 'B', 16);
        // mencetak string
        if($start == " AND $end == ") {
            $pdf->Cell(330,7,$title,0,1,'C');
            $pdf->SetFont('Arial', 'B', 14);
            $pdf->Cell(330,7,$atr['nama'],0,1,'C');
            $pdf->SetFont('Arial', 'B', 12);
            $pdf->Cell(330,7,$atr['alamat'],0,1,'C');
        }else {
            $pdf->Cell(330,7,$title,0,1,'C');
            $pdf->SetFont('Arial', 'B', 14);
            $pdf->Cell(330,7,'Periode '.date('d-m-
Y', strtotime($start)).' S/D '.date('d-m-Y',
strtotime($end)),0,1,'C');
            $pdf->SetFont('Arial', 'B', 14);
            $pdf->Cell(330,7,$atr['nama'],0,1,'C');
            $pdf->SetFont('Arial', 'B', 12);
            $pdf->Cell(330,7,$atr['alamat'],0,1,'C');
        }
    }
}

```

```

    }
    $pdf->Ln(6);
    $pdf->SetFont("Times","B",11);
    $pdf->Cell(15,8,'No',1,0,'C',1);
    $pdf->Cell(35,8,'Tanggal',1,0,'C',1);
    $pdf->Cell(120,8,'Barang',1,0,'C',1);
    $pdf->Cell(35,8,'Permintaan Stok',1,0,'C',1);
    $pdf->Cell(80,8,'Supplier',1,0,'C',1);
    $pdf->Cell(50,8,'Status',1,1,'C',1);

    $pdf->SetFont("Times","",10);
    $i = 1;
    foreach($pemesanan as $psn) {
        $pdf->Cell(15,8,$i.',',1,0,'T');
        $pdf->Cell(35,8,date('d/m/Y',
strtotime($psn['pemesn_tgl']),1,0,'T');
        $pdf-
>Cell(120,8,$psn['barang_nama'],1,0,'T');
        $pdf-
>Cell(35,8,number_format($psn['pemesn_stok'],
0,',','.').'$psn['satuan_nama'],1,0,'T');
        $pdf-
>Cell(80,8,$psn['supplier_nama'],1,0,'T');
        $pdf-
>Cell(50,8,$psn['pemesn_status'],1,1,'T');
        $i++;
    }
    $pdf->Output('D',$title.'.pdf');
}

public function konfigurasi_atribut() {
    $data = array (
        'title'
=>    'Atribut',
        'me'
=>    $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        'atr'
=>    $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
    );
    $this->form_validation-
>set_rules('nama', 'nama', 'required', [
        'required'=>    'Kolom ini tidak
boleh kosong']);
    $this->form_validation-
>set_rules('alamat', 'alamat', 'required', [
        'required'=>    'Kolom ini tidak
boleh kosong'];
    if($this->form_validation-
>run() == FALSE) {
        $this->load-
>view('admin/header', $data);

```

```

        $this->load-
>view('admin/atribut', $data);
        $this->load-
>view('admin/footer');
    }else {
        $this-
>Admin_model->simpan_atribut();
        $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

        redirect('admin/konfigurasi/atribut');
    }
}

public function konfigurasi_favicon() {
    $data = array (
        'title'
=>    'Favicon',
        'me'
=>    $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        'atr'
=>    $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
    );
    $this->form_validation-
>set_rules('id', 'id', 'required', [
        'required'=>    'Kolom ini tidak
boleh kosong'];
    if($this->form_validation-
>run() == FALSE) {
        $this->load-
>view('admin/header', $data);
        $this->load-
>view('admin/favicon', $data);
        $this->load-
>view('admin/footer');
    }else {
        $this-
>Admin_model->simpan_favicon();
        $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

        redirect('admin/konfigurasi/favicon');
    }
}

public function konfigurasi_batas() {
    $data = array (
        'title'
=>    'Batas Stok Barang',
        'me'

```

```

=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
    'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
    );
    $this->form_validation-
>set_rules('min', 'min', 'required', [
    'required'=> 'Kolom ini tidak
    boleh kosong']);
    $this->form_validation-
>set_rules('max', 'max', 'required', [
    'required'=> 'Kolom ini tidak
    boleh kosong']);
    if($this->form_validation-
>run() == FALSE) {
        $this->load-
>view('admin/header', $data);
        $this->load-
>view('admin/batas_minimal', $data);
        $this->load-
>view('admin/footer');
    }else {
        $this->db-
>update('tb_atribut',['batas' => $this->input-
>post('min'), 'max' => $this->input->post('max')],
array('id' => 1));
        $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

        redirect('admin/konfigurasi/batas-stok');
    }
}

public function konfigurasi_wapi() {
    $data = array (
        'title'
=> 'API WhatsApp',
        'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
    );
    $this->form_validation-
>set_rules('token', 'token', 'required', [
    'required'=> 'Kolom ini tidak
    boleh kosong']);
    $this->form_validation-

```

```

>set_rules('url', 'url', 'required', [
    'required'=> 'Kolom ini tidak
    boleh kosong']);
    if($this->form_validation-
>run() == FALSE) {
        $this->load-
>view('admin/header', $data);
        $this->load-
>view('admin/wapi', $data);
        $this->load-
>view('admin/footer');
    }else {
        $this->db-
>update('tb_atribut',['token' => $this->input-
>post('token'),'url' => $this->input->post('url')],
array('id' => 1));
        $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

        redirect('admin/konfigurasi/api-
whatsapp');
    }
}

public function read_notif($id) {
    $this->db->set('notif_status',
'Dibaca');
    $this->db->where('notif_id',
$id);
    $this->db-
>update('tb_notifikasi');
    redirect('admin/daftar-
pemesanan');
}

public function profil() {
    $data = array (
        'title'
=> 'Profil',
        'me'
=> $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
    );
    $this->form_validation-
>set_rules('nama', 'nama', 'required', [
    'required'=> 'Kolom ini tidak
    boleh kosong']);
    $this->form_validation-
>set_rules('username', 'username', 'required', [

```

```

        'required'=>      'Kolom ini tidak
boleh kosong'];
        $this->form_validation-
>set_rules('token', 'token', 'required', [

        'required'=>      'Kolom ini tidak
boleh kosong'];
        $this->form_validation-
>set_rules('password', 'password', 'required', [

        'required'=>      'Kolom ini tidak
boleh kosong'];
        if($this->form_validation-
>run() == FALSE) {
                $this->load-
>view('admin/header', $data);
                $this->load-
>view('admin/profil', $data);
                $this->load-
>view('admin/footer', $data);
        }else {
                $this-
>Admin_model->simpan_profil();
        }
}

public function password() {
    $data = array (
        'title'
=>      'Password',
        'me'
=>      $this->db-
>get_where('tb_users',['user_id' => $this-
>session->userdata('id')])>row_array(),
        'atr'
=>      $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
    );
    $this->form_validation-
>set_rules('password1', 'password1', 'required', [

        'required'=>      'Kolom ini tidak
boleh kosong'];
        $this->form_validation-
>set_rules('password2', 'password2',
'required|matches[password1]', [

        'required'=>      'Kolom ini tidak
boleh kosong',

        'matches'=>      'Konfirmasi
password harus sama']);
        $this->form_validation-
>set_rules('password', 'password', 'required', [

```

```

        'required'=>      'Kolom ini tidak
boleh kosong'];
        if($this->form_validation-
>run() == FALSE) {
                $this->load-
>view('admin/header', $data);
                $this->load-
>view('admin/password', $data);
                $this->load-
>view('admin/footer', $data);
        }else {
                $this-
>Admin_model->simpan_password();
        }
}

/* End of file Admin.php */
/* Location: ./application/controllers/Admin.php
*/

```

### 3. Supplier.php

```

<?php
defined('BASEPATH') OR exit('No direct script
access allowed');

class Supplier extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load-
>model('Supplier_model');
        if($this->session-
>userdata('status') != 'login_supplier') {
            redirect("");
        }
    }

    public function dashboard() {
        $data = array (
            'title'
=>      'Dashboard',
            'me'
=>      $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
=>      $this->db-
>get_where('tb_atribut',['id' => 1])>row_array(),
            'kategori'
=>      $this->db-
>get_where('tb_kategori',['kat_user' => $this-
>session->userdata('id')])>num_rows(),
            'barang'
=>      $this->db-

```

```

>get_where('tb_barang',['barang_user' => $this-
>session->userdata('id')])->num_rows(),
        'request'
=> $this->db-
>get_where('tb_pemesanan',['pemesn_supplier'
=> $this->session->userdata('id')])->num_rows(),
        'penawaran'
=> $this->db-
>get_where('tb_penawaran',['penawaran_supplier
' => $this->session->userdata('id')])->
num_rows(),
        'grafikpesanan'
=> $this->Supplier_model-
>grafik_pesananku(),
        'grafikkategori'=>
        $this->Supplier_model-
>grafik_kategori(),
    );
    $this->load-
>view('supplier/header', $data);
    $this->load-
>view('supplier/index', $data);
    $this->load-
>view('supplier/footer');
    }

    public function kategori() {
        $data = array (
            'title'
=> 'Kategori',
            'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'kategori'
=> $this->Supplier_model-
>data_kategori(),
        );
        $this->load-
>view('supplier/header', $data);
        $this->load-
>view('supplier/kategori', $data);
        $this->load-
>view('supplier/footer');
    }

    public function kategori_add() {
        $data = array (
            'title'
=> 'Input Kategori',
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),

```

```

        'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        );
        $this->form_validation-
>set_rules('nama', 'nama', 'required', [
            'required'=> 'Kolom ini tidak
boleh kosong']);
        if($this->form_validation-
>run() == FALSE) {
            $this->load-
>view('supplier/header', $data);
            $this->load-
>view('supplier/kategori_add', $data);
            $this->load-
>view('supplier/footer');
        }else {
            $this-
>Supplier_model->simpan_kategori();
            $this->session-
>set_flashdata('flash', 'Data baru berhasil
ditambahkan');

            redirect('supplier/master/kategori');
        }

        public function kategori_edit($id) {
            $data = array (
                'title'
=> 'Edit Kategori',
                'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
                'katid'
=> $this->db-
>get_where('tb_kategori',['kat_id' => $id])->
row_array(),
            );
            $this->form_validation-
>set_rules('nama', 'nama', 'required', [
                'required'=> 'Kolom ini tidak
boleh kosong']);
            if($this->form_validation-
>run() == FALSE) {
                $this->load-
>view('supplier/header', $data);
                $this->load-
>view('supplier/kategori_edit', $data);

```

```

        $this->load-
>view('supplier/footer');
        }else {
            $this-
>Supplier_model->edit_kategori($id);
            $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

            redirect('supplier/master/kategori');
        }
        public function kategori_del($id) {
            $this->db->where('kat_id',
            $id);
            $this->db-
>delete('tb_kategori');
            $this->session-
>set_flashdata('flash', 'Data berhasil dihapus');

            redirect('supplier/master/kategori');
        }

        public function kategori_kosongkan() {
            $this->db->where('kat_user',
            $this->session->userdata('id'));
            $this->db-
>delete('tb_kategori');
            $this->session-
>set_flashdata('flash', 'Tabel berhasil
dikosongkan');

            redirect('supplier/master/kategori');
        }

        public function barang() {
            $data = array (
                'title'
            => 'Barang',
                'me'
            => $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
                'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'barang'
            => $this->Supplier_model-
>data_barang(),
        );
        $this->load-
>view('supplier/header', $data);
        $this->load-
>view('supplier/barang', $data);
        $this->load-
>view('supplier/footer');

```

```

        }
        public function barang_add() {
            $data = array (
                'title'
            => 'Input Barang',
                'atr'
            => $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'me'
            => $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
                'kategori'
            => $this->Supplier_model-
>data_kategori(),
                'satuan'
            => $this->db->get('tb_satuan')-
>result_array(),
        );
        $this->form_validation-
>set_rules('id', 'id',
'required|is_unique[tb_barang.barang_id]', [
                'required'=>
                'Kolom ini tidak boleh kosong',
                'is_unique' => 'Id barang
ini sudah digunakan']);
        $this->form_validation-
>set_rules('nama', 'nama', 'required', [
                'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation-
>set_rules('kategori', 'kategori', 'required', [
                'required'=> 'Kolom ini tidak
boleh kosong']);
        $this->form_validation-
>set_rules('harga', 'harga', 'required', [
                'required'=>
                'Kolom ini tidak boleh kosong']);
        $this->form_validation-
>set_rules('stok', 'stok', 'required', [
                'required'=>
                'Kolom ini tidak boleh kosong']);
        $this->form_validation-
>set_rules('satuan', 'satuan', 'required', [
                'required'=> 'Kolom ini tidak
boleh kosong']);
        if($this->form_validation-
>run() == FALSE) {
            $this->load-
>view('supplier/header', $data);
            $this->load-
>view('supplier/barang_add', $data);

```

```

        $this->load-
>view('supplier/footer');
        }else {
            $this-
>Supplier_model->simpan_barang();
            $this->session-
>set_flashdata('flash', 'Data baru berhasil
ditambahkan');

        redirect('supplier/master/barang');
        }
        public function barang_edit($id) {
            $data = array (
                'title'
=>        'Edit Barang',
                'atr'
=>        $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
                'me'
=>        $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
                'kategori'
=>        $this->Supplier_model-
>data_kategori(),
                'barid'
=>        $this->db-
>get_where('tb_barang',['barang_id' => $id])-
>row_array(),
                'satuan'
=>        $this->db->get('tb_satuan')-
>result_array(),
            );
            $this->form_validation-
>set_rules('id', 'id', 'required', [
                'required'=>
                'Kolom ini tidak boleh kosong']);
            $this->form_validation-
>set_rules('nama', 'nama', 'required', [
                'required'=>
                'Kolom ini tidak boleh kosong']);
            $this->form_validation-
>set_rules('kategori', 'kategori', 'required', [
                'required'=>        'Kolom ini tidak
                boleh kosong']);
            $this->form_validation-
>set_rules('harga', 'harga', 'required', [
                'required'=>        'Kolom ini tidak
                boleh kosong']);
            $this->form_validation-
>set_rules('stok', 'stok', 'required', [
                'required'=>
                'Kolom ini tidak boleh kosong']);

```

```

        $this->form_validation-
>set_rules('satuan', 'satuan', 'required', [
                'required'=>        'Kolom ini tidak
                boleh kosong']);
            if($this->form_validation-
>run() == FALSE) {
                $this->load-
>view('supplier/header', $data);
                $this->load-
>view('supplier/barang_edit', $data);
                $this->load-
>view('supplier/footer');
            }else {
                $this-
>Supplier_model->edit_barang($id);
                $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

                redirect('supplier/master/barang');
            }
            public function barang_del($id) {
                $this->db->where('barang_id',
                $id);
                $this->db-
>delete('tb_barang');
                $this->session-
>set_flashdata('flash', 'Data berhasil dihapus');
                redirect('supplier/master/barang');
            }
            public function barang_kosongkan() {
                $this->db-
>where('barang_user', $this->session-
>userdata('id'));
                $this->db-
>delete('tb_barang');
                $this->session-
>set_flashdata('flash', 'Tabel berhasil
dikosongkan');

                redirect('supplier/master/barang');
            }
            public function barang_offer($id) {
                $cek = $this->db-
>get_where('tb_barang',['barang_id' => $id])-
>row_array();
                $data = array (
                    'penawaran_id'
=>        $cek['barang_id'],
                    'penawaran_kategori'
=>        $cek['barang_kategori'],
                    'penawaran_nama'
=>        $cek['barang_nama'],
                    'penawaran_harga'

```

```

=> $cek['barang_harga'],
    'penawaran_stok'
=> $cek['barang_stok'],
    'penawaran_supplier'
=> $cek['barang_user'],
    'penawaran_satuan'
    => $cek['barang_satuan'],
    'penawaran_foto'
=> $cek['barang_foto'],
    );
    $this->db-
>insert('tb_penawaran', $data);
    $this->session-
>set_flashdata('flash', 'Barang berhasil
ditambahkan sebagai penawaran');
    redirect('supplier/penawaran-
barang');
}
public function list_pemesanan() {
    $data = array (
        'title'
=> 'Daftar Pesanan',
        'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'pemesanan'
=> $this->Supplier_model-
>data_pemesanan(),
    );
    $this->load-
>view('supplier/header', $data);
    $this->load-
>view('supplier/pemesanan', $data);
    $this->load-
>view('supplier/footer');
}
public function konfirmasi_barang($id)
{
    $this->session-
>set_userdata('idpemesanan', $id);
    $cek = $this->db-
>get_where('tb_pemesanan',['pemesn_id' =>
$id])->row_array();
    $cekbrg = $this->db-
>get_where('tb_barang',['barang_id' =>
$cek['pemesn_barang']])->row_array();
    $key =
$cekbrg['barang_nama'];
    $data = array (
        'title'
=> 'Pilih Stok Barang',
        'me'

```

```

=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'brg'
=> $this->Supplier_model-
>data_barang_sortir($key),
    );
    $this->load-
>view('supplier/header', $data);
    $this->load-
>view('supplier/list_barang', $data);
    $this->load-
>view('supplier/footer');
}
public function
konfirmasi_barang_pilih($id) {
    $data = array (
        'title'
=> 'Konfirmasi Pesanan Stok',
        'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'barid'
=> $this->db-
>get_where('tb_barang',['barang_id' => $id])->
row_array(),
        'pmsid'
=> $this->db-
>get_where('tb_pemesanan',['pemesn_id' =>
$this->session->userdata('idpemesanan')])->
row_array(),
    );
    $this->form_validation-
>set_rules('barang', 'barang', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong'];
    $this->form_validation-
>set_rules('stok', 'stok', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong'];
    if($this->form_validation-
>run() == FALSE) {
        $this->load-
>view('supplier/header', $data);
        $this->load-
>view('supplier/konfirmasi_pesanan', $data);
        $this->load-

```

```

>view('supplier/footer');
    }else {
        $this->Supplier_model->terima_pesanan($id);
        $this->session->unset_userdata('idpemesanan');
        $this->session->set_flashdata('flash', 'Permintaan stok diterima');

        redirect('supplier/pesanan-toko');
    }
    public function list_pemesanan_tolak($id) {
        $data = array (
            'title' => 'Beri Alasan',
            'atr' => $this->db->get_where('tb_atribut',['id' => 1])->row_array(),
            'me' => $this->db->get_where('tb_supplier',['supplier_id' => $this->session->userdata('id')])->row_array(),
            'pmsid' => $this->db->get_where('tb_pemesanan',['pemesn_id' => $id])->row_array(),
        );
        $this->form_validation->set_rules('isi', 'isi', 'required', [
            'required' => 'Kolom ini tidak boleh kosong']);
        if($this->form_validation->run() == FALSE) {
            $this->load->view('supplier/header', $data);
            $this->load->view('supplier/tolak_request', $data);
            $this->load->view('supplier/footer');
        }else {
            $this->Supplier_model->simpan_tanggapan($id);
            $this->session->set_flashdata('flash', 'Tanggapan anda berhasil dikirim');

            redirect('supplier/pesanan-toko');
        }
        public function penawaran() {
            $data = array (
                'title' => 'Penawaran Barang',
                'me' => $this->db->get_where('tb_supplier',['supplier_id' => $this->session->userdata('id')])->row_array(),
                'atr' => $this->db->get_where('tb_atribut',['id' => 1])->row_array(),
                'barang' => $this->Supplier_model->data_penawaran(),
            );
            $this->load->view('supplier/header', $data);
            $this->load->view('supplier/penawaran', $data);
            $this->load->view('supplier/footer');
        }

        public function penawaran_edit($id) {
            $data = array (
                'title' => 'Edit Penawaran Barang',
                'atr' => $this->db->get_where('tb_atribut',['id' => 1])->row_array(),
                'me' => $this->db->get_where('tb_supplier',['supplier_id' => $this->session->userdata('id')])->row_array(),
                'kategori' => $this->Supplier_model->data_kategori(),
                'barid' => $this->db->get_where('tb_penawaran',['penawaran_id' => $id])->row_array(),
                'satuan' => $this->db->get('tb_satuan')->result_array(),
            );
            $this->form_validation->set_rules('nama', 'nama', 'required', [
                'required' => 'Kolom ini tidak boleh kosong']);
            $this->form_validation->set_rules('kategori', 'kategori', 'required', [
                'required' => 'Kolom ini tidak boleh kosong']);
            $this->form_validation->set_rules('harga', 'harga', 'required', [
                'required' => 'Kolom ini tidak boleh kosong']);
        }
    }
}

```

```

                $this->form_validation-
>set_rules('stok', 'stok', 'required', [
                'required'=>    'Kolom ini
tidak boleh kosong']);
                $this->form_validation-
>set_rules('satuan', 'satuan', 'required', [
                'required'=>    'Kolom ini tidak
boleh kosong']);
                if($this->form_validation-
>run() == FALSE) {
                        $this->load-
>view('supplier/header', $data);
                        $this->load-
>view('supplier/penawaran_edit', $data);
                        $this->load-
>view('supplier/footer');
                }else {
                        $this-
>Supplier_model->edit_penawaran($id);
                        $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');

                        redirect('supplier/penawaran-barang');
                }
                public function penawaran_del($id) {
                        $this->db-
>where('penawaran_id', $id);
                        $this->db-
>delete('tb_penawaran');
                        $this->session-
>set_flashdata('flash', 'Data berhasil dihapus');
                        redirect('supplier/penawaran-
barang');
                }
                public function
penawaran_kosongkan() {
                        $this->db-
>where('penawaran_supplier', $this->session-
>userdata('id'));
                        $this->db-
>delete('tb_penawaran');
                        $this->session-
>set_flashdata('flash', 'Tabel berhasil
dikosongkan');
                        redirect('supplier/penawaran-barang');
                }
                public function penawaran_stok($id) {
                        $cek = $this->db-
>get_where('tb_barang', ['barang_id' => $id])-
>row_array();
                        $this->db-
>set('penawaran_stok', $cek['barang_stok']);
                        $this->db-

```

```

>set('penawaran_harga', $cek['barang_harga']);
                        $this->db-
>where('penawaran_id', $id);
                        $this->db-
>update('tb_penawaran');
                        $this->session-
>set_flashdata('flash', 'Data berhasil
diperbaharui');
                        redirect('supplier/penawaran-
barang');
                }
                public function barang_mitra() {
                        $data = array (
                                'title'
=>    'Stok Barang Mitra',
                                'me'
=>    $this->db-
>get_where('tb_supplier', ['supplier_id' => $this-
>session->userdata('id')])->row_array(),
                                'atr'
=>    $this->db-
>get_where('tb_atribut', ['id' => 1])->row_array(),
                                'barang'
=>    $this->Supplier_model-
>barang_mitra(),
                                );
                        $this->load-
>view('supplier/header', $data);
                        $this->load-
>view('supplier/barang_mitra', $data);
                        $this->load-
>view('supplier/footer');
                }
                public function list_pesanan() {
                        $data = array (
                                'title'
=>    'Data Pesanan Barang',
                                'me'
=>    $this->db-
>get_where('tb_supplier', ['supplier_id' => $this-
>session->userdata('id')])->row_array(),
                                'atr'
=>    $this->db-
>get_where('tb_atribut', ['id' => 1])->row_array(),
                                'barang'
=>    $this->Supplier_model-
>data_pesanan(),
                                );
                        $this->load-
>view('supplier/header', $data);
                        $this->load-
>view('supplier/pesanan_barang', $data);
                        $this->load-
>view('supplier/footer');
                }
                public function pesbar_konfirmasi() {

```

```

        $this->db-
>update('tb_pesanan_barang',['pesbar_stok' =>
$this->input->post('jml'),'pesbar_harga' => $this-
>input->post('harga'),'pesbar_catatan' => $this-
>input->post('catatan'),'pesbar_status' =>
'Dikonfirmasi'],array('pesbar_id' => $this->input-
>post('id')));
        $this->session-
>set_flashdata('flash', 'Pesanan barang toko
dikonfirmasi');
        redirect('supplier/pesan-
barang');
    }
    public function pesbar_alasan() {
        $this->db-
>update('tb_pesanan_barang',['pesbar_status' =>
'Ditolak','pesbar_catatan' => $this->input-
>post('alasan')],array('pesbar_id' => $this->input-
>post('id')));
        $this->session-
>set_flashdata('flash', 'Pesanan barang toko
ditolak');
        redirect('supplier/pesan-
barang');
    }
    public function lap_kategori() {
        $data = array (
            'title'
=> 'Laporan Data Kategori',
            'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'kategori'
=> $this->Supplier_model-
>data_kategori(),
        );
        $this->load-
>view('supplier/header', $data);
        $this->load-
>view('supplier/lap_kategori', $data);
        $this->load-
>view('supplier/footer');
    }
    public function lap_kategori_print() {
        $data = array (
            'title'
=> 'Laporan Data Kategori',
            'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'

```

```

=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'kategori'
=> $this->Supplier_model-
>data_kategori(),
        );
        $this->load-
>view('supplier/lap_kategori_print', $data);
    }
    public function lap_kategori_excel() {
        $data = array (
            'title'
=> 'Laporan Data Kategori',
            'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'kategori'
=> $this->Supplier_model-
>data_kategori(),
        );
        $this->load-
>view('supplier/lap_kategori_excel', $data);
    }
    public function lap_kategori_pdf() {
        $title = 'Laporan Data
Kategori';
        $me = $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array();
        $kategori = $this-
>Supplier_model->data_kategori();
        $pdf = new
FPDF('p','mm','Legal');
        // membuat halaman baru
        $pdf->AddPage();
        // setting jenis font yang akan digunakan
        $pdf->SetFont('Arial', 'B', 12);
        $pdf-
>Image(base_url(). 'assets/admin/images/'. $me['s
upplier_foto'], 30, 12, -300);
        $pdf->SetFont('Arial', 'B', 16);
        // mencetak string
        $pdf->Cell(195,7,$title,0,1,'C');
        $pdf->SetFont('Arial', 'B', 14);
        $pdf-
>Cell(195,7,$me['supplier_nama'],0,1,'C');
        $pdf->SetFont('Arial', 'B', 12);
        $pdf-
>Cell(195,7,$me['supplier_alamat'],0,1,'C');
        $pdf->Ln(6);
        $pdf->SetFont('Times', 'B', 11);
        $pdf->Cell(10,8,'No',1,0,'C',1);
    }

```

```

$pdf->Cell(96,8,'Kategori',1,0,'C',1);
$pdf->Cell(90,8,'Jumlah Barang',1,1,'C',1);
$pdf->SetFont('Times','',10);
$i = 1;
    foreach($kategori as $ktg) {
        $countitem = $this->db-
>get_where('tb_barang',['barang_kategori' =>
$ktg['kat_id']])->num_rows();
        $pdf->Cell(10,8,$i.'',1,0,'T');
        $pdf-
>Cell(96,8,$ktg['kat_nama'],1,0,'T');
        $pdf-
>Cell(90,8,$countitem,1,1,'T');
        $i++;
    }
$pdf->Output('D',$title.'.pdf');
}

public function lap_barang() {
    $data = array (
        'title'
=>    'Laporan Data Barang',
        'me'
=>    $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=>    $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'barang'
=>    $this->Supplier_model-
>data_barang(),
    );
    $this->load-
>view('supplier/header', $data);
    $this->load-
>view('supplier/lap_barang', $data);
    $this->load-
>view('supplier/footer');
}

public function lap_barang_print() {
    $data = array (
        'title'
=>    'Laporan Data Barang',
        'me'
=>    $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=>    $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'barang'
=>    $this->Supplier_model-
>data_barang(),
    );
    $this->load-

```

```

>view('supplier/lap_barang_print', $data);
    }
    public function lap_barang_excel() {
        $data = array (
            'title'
=>    'Laporan Data Barang',
            'me'
=>    $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
=>    $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
            'barang'
=>    $this->Supplier_model-
>data_barang(),
        );
        $this->load-
>view('supplier/lap_barang_excel', $data);
    }

    public function lap_barang_pdf() {
        $title = 'Laporan Data
Barang';
        $me = $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array();
        $barang = $this-
>Supplier_model->data_barang();
        $pdf = new
FPDF('l','mm','Legal');
        // membuat halaman baru
        $pdf->AddPage();
        // setting jenis font yang akan digunakan
        $pdf->SetFont('Arial','B',16);
        $pdf-
>Image(base_url().'assets/admin/images/'.$me['s
upplier_foto'], 30, 12, -300);
        $pdf->SetFont('Arial','B',16);
        // mencetak string
        $pdf->Cell(330,7,$title,0,1,'C');
        $pdf->SetFont('Arial','B',14);
        $pdf-
>Cell(330,7,$me['supplier_nama'],0,1,'C');
        $pdf->SetFont('Arial','B',12);
        $pdf-
>Cell(330,7,$me['supplier_alamat'],0,1,'C');
        $pdf->Ln(6);
        $pdf->SetFont('Times','B',11);
        $pdf->Cell(15,8,'No',1,0,'C',1);
        $pdf->Cell(170,8,'Nama',1,0,'C',1);
        $pdf->Cell(55,8,'Kategori',1,0,'C',1);
        $pdf->Cell(60,8,'Harga',1,0,'C',1);
        $pdf->Cell(35,8,'Stok',1,1,'C',1);

        $pdf->SetFont('Times','',10);
        $i = 1;

```



```

        $pdf->Cell(10,8,$i.',1,0,1');
        $pdf-
>Cell(116,8,$brg['barang_nama'],1,0,1');
        $pdf-
>Cell(70,8,number_format($brg['barang_stok'],0
,','.'.'.'.$brg['satuan_nama'],1,1,1');
        $i++;
    }
    $pdf->Output('D',$title.'.pdf');
}
public function lap_penawaran() {
    $data = array (
        'title'
=> 'Laporan Penawaran Barang',
        'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'barang'
=> $this->Supplier_model-
>data_penawaran(),
    );
    $this->load-
>view('supplier/header', $data);
    $this->load-
>view('supplier/lap_penawaran', $data);
    $this->load-
>view('supplier/footer');
}

public function lap_penawaran_print()
{
    $data = array (
        'title'
=> 'Laporan Penawaran Barang',
        'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'barang'
=> $this->Supplier_model-
>data_penawaran(),
    );
    $this->load-
>view('supplier/lap_penawaran_print', $data);
}
public function lap_penawaran_excel()
{
    $data = array (
        'title'
=> 'Laporan Penawaran Barang',
        'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'barang'
=> $this->Supplier_model-
>data_penawaran(),
    );
    $this->load-
>view('supplier/lap_penawaran_excel', $data);
}
}

public function lap_penawaran_pdf() {
    $title = 'Laporan Penawaran
Barang';
    $me = $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array();
    $barang = $this-
>Supplier_model->data_penawaran();
    $pdf = new
FPDF('l','mm','Legal');
    // membuat halaman baru
    $pdf->AddPage();
    // setting jenis font yang akan digunakan
    $pdf->SetFillColor(112, 226, 137);
    $pdf-
>Image(base_url(). 'assets/admin/images/'. $me['s
upplier_foto'], 30, 12, -300);
    $pdf->SetFont('Arial','B',16);
    // mencetak string
    $pdf->Cell(330,7,$title,0,1,'C');
    $pdf->SetFont('Arial','B',14);
    $pdf-
>Cell(330,7,$me['supplier_nama'],0,1,'C');
    $pdf->SetFont('Arial','B',12);
    $pdf-
>Cell(330,7,$me['supplier_alamat'],0,1,'C');
    $pdf->Ln(6);
    $pdf->SetFont('Times','B',11);
    $pdf->Cell(15,8,'No',1,0,'C',1);
    $pdf->Cell(170,8,'Nama',1,0,'C',1);
    $pdf->Cell(55,8,'Kategori',1,0,'C',1);
    $pdf->Cell(60,8,'Harga',1,0,'C',1);
    $pdf->Cell(35,8,'Stok',1,1,'C',1);

    $pdf->SetFont('Times','',10);
    $i = 1;
    foreach($barang as $brg) {
        $pdf->Cell(15,8,$i.',1,0,1');
        $pdf-
>Cell(170,8,$brg['penawaran_nama'],1,0,1');

```

```

        $pdf-
>Cell(55,8,$brg['kat_nama'],1,0,'l');
        $pdf-
>Cell(60,8,number_format($brg['penawaran_har
ga'],0,',',''),1,0,'l');
        $pdf-
>Cell(35,8,number_format($brg['penawaran_sto
k'],0,',','').$brg['satuan_nama'],1,1,'l');
        $i++;
    }
    $pdf->Output('D', $title.'.pdf');
}

public function lap_pestok() {
    $start = $this->input-
>post('start');
    $end = $this->input-
>post('end');
    if($start == " AND $end == ")
    {
        $pemesanan = $this-
>Supplier_model->data_pemesanan();
    }else {
        $pemesanan = $this-
>Supplier_model-
>data_pemesanan_sortir(array($start,$end));
    }
    $data = array (
        'title'
=> 'Laporan Daftar Pesanan
Toko',
        'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'pemesanan'
=> $pemesanan,
        'start'
=> $start,
        'end'
=> $end,
    );
    $this->load-
>view('supplier/header', $data);
    $this->load-
>view('supplier/lap_pemesanan', $data);
    $this->load-
>view('supplier/footer');
}

public function lap_pestok_print() {
    $start = $this->uri-
>segment(5);

```

```

    $end = $this->uri-
>segment(6);
    if($start == " AND $end == ")
    {
        $pemesanan = $this-
>Supplier_model->data_pemesanan();
    }else {
        $pemesanan = $this-
>Supplier_model-
>data_pemesanan_sortir(array($start,$end));
    }
    $data = array (
        'title'
=> 'Laporan Daftar Pesanan
Toko',
        'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
        'atr'
=> $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),
        'pemesanan'
=> $pemesanan,
        'start'
=> $start,
        'end'
=> $end,
    );
    $this->load-
>view('supplier/lap_pemesanan_print', $data);
}
    public function lap_pestok_excel() {
        $start = $this->uri-
>segment(5);
        $end = $this->uri-
>segment(6);
        if($start == " AND $end == ")
        {
            $pemesanan = $this-
>Supplier_model->data_pemesanan();
        }else {
            $pemesanan = $this-
>Supplier_model-
>data_pemesanan_sortir(array($start,$end));
        }
        $data = array (
            'title'
=> 'Laporan Daftar Pesanan
Toko',
            'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
=> $this->db-

```

```

>get_where('tb_atribut',['id' => 1])->row_array(),
    'pemesanan'
=>    $pemesanan,
    'start'
=>    $start,
    'end'
=>    $end,
    );
    $this->load-
>view('supplier/lap_pemesanan_excel', $data);
    }
    public function lap_pestok_pdf() {
        $start = $this->uri-
>segment(5);
        $end = $this->uri-
>segment(6);
        if($start == " AND $end == ")
        {
            $pemesanan = $this-
>Supplier_model->data_pemesanan();
        }else {
            $pemesanan = $this-
>Supplier_model-
>data_pemesanan_sortir(array($start,$end));
        }
        $title = 'Laporan Daftar
Pesanan Toko';
        $me = $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array();
        $pdf = new
FPDF('T','mm','Legal');
        // membuat halaman baru
        $pdf->AddPage();
        // setting jenis font yang akan digunakan
        $pdf->SetFont('Arial', 'B', 12);
        $pdf-
>Image(base_url().'/assets/admin/images/'.$me['s
upplier_foto'], 30, 12, -300);
        $pdf->SetFont('Arial','B',16);
        // mencetak string
        if($start == " AND $end == ") {
            $pdf->Cell(330,7,$title,0,1,'C');
            $pdf->SetFont('Arial','B',14);
            $pdf-
>Cell(330,7,$me['supplier_nama'],0,1,'C');
            $pdf->SetFont('Arial','B',12);
            $pdf-
>Cell(330,7,$me['supplier_alamat'],0,1,'C');
        }else {
            $pdf->Cell(330,7,$title,0,1,'C');
            $pdf->SetFont('Arial','B',14);
            $pdf->Cell(330,7,Periode
            '.date('d-m-Y', strtotime($start)).' S/D '.date('d-m-
            Y', strtotime($end)),0,1,'C');
            $pdf->SetFont('Arial','B',14);

```

```

        $pdf-
>Cell(330,7,$me['supplier_nama'],0,1,'C');
        $pdf->SetFont('Arial','B',12);
        $pdf-
>Cell(330,7,$me['supplier_alamat'],0,1,'C');
    }
    $pdf->Ln(6);
    $pdf->SetFont('Times','B',11);
    $pdf->Cell(15,8,'No',1,0,'C',1);
    $pdf->Cell(35,8,'Tanggal',1,0,'C',1);
    $pdf->Cell(120,8,'Barang',1,0,'C',1);
    $pdf->Cell(35,8,'Permintaan Stok',1,0,'C',1);
    $pdf->Cell(80,8,'Dari Toko',1,0,'C',1);
    $pdf->Cell(50,8,'Status',1,1,'C',1);

    $pdf->SetFont('Times','',10);
    $i = 1;
    foreach($pemesanan as $psn) {
        $pdf->Cell(15,8,$i.',',1,0,'T');
        $pdf->Cell(35,8,date('d/m/Y',
strtotime($psn['pemesn_tgl']),1,0,'T');
        $pdf-
>Cell(120,8,$psn['barang_nama'],1,0,'T');
        $pdf-
>Cell(35,8,number_format($psn['pemesn_stok'],
0,',','.').'. $psn['satuan_nama'],1,0,'T');
        $pdf-
>Cell(80,8,$psn['user_nama'],1,0,'T');
        $pdf-
>Cell(50,8,$psn['pemesn_status'],1,1,'T');
        $i++;
    }
    $pdf->Output('D',$title.'.pdf');
    }

    public function read_notif($id) {
        $this->db->set('notif_status',
'Dibaca');
        $this->db-
>wh49*ere('notif_id', $id);
        $this->db-
>update('tb_notifikasi');
        redirect('supplier/pesanan-
toko');
    }

    public function profil() {
        $data = array (
            'title'
=>        'Profil',
            'me'
=>        $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])->row_array(),
            'atr'
=>        $this->db-
>get_where('tb_atribut',['id' => 1])->row_array(),

```

```

    );
    $this->form_validation-
>set_rules('nama', 'nama', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong'];
    $this->form_validation-
>set_rules('alamat', 'alamat', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong'];
    $this->form_validation-
>set_rules('nohp', 'nohp', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong'];
    $this->form_validation-
>set_rules('username', 'username', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong'];
    $this->form_validation-
>set_rules('token', 'token', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong'];
    $this->form_validation-
>set_rules('password', 'password', 'required', [
        'required'=> 'Kolom ini tidak
boleh kosong'];
    if($this->form_validation-
>run() == FALSE) {
        $this->load-
>view('supplier/header', $data);
        $this->load-
>view('supplier/profil', $data);
        $this->load-
>view('supplier/footer', $data);
    }else {
        $this-
>Supplier_model->simpan_profil();
    }
    public function password() {
        $data = array (
            'title'
=> 'Password',
            'me'
=> $this->db-
>get_where('tb_supplier',['supplier_id' => $this-
>session->userdata('id')])>row_array(),
            'atr'
=> $this->db-
>get_where('tb_tribut',['id' => 1])>row_array(),
        );

```

```

        $this->form_validation-
>set_rules('password1', 'password1', 'required', [
            'required'=>
'Kolom ini tidak boleh kosong'];
        $this->form_validation-
>set_rules('password2', 'password2',
'required|matches[password1]', [
            'required'=> 'Kolom ini tidak
boleh kosong',
            'matches'=> 'Konfirmasi
password harus sama'];
        $this->form_validation-
>set_rules('password', 'password', 'required', [
            'required'=> 'Kolom ini tidak
boleh kosong'];
        if($this->form_validation-
>run() == FALSE) {
            $this->load-
>view('supplier/header', $data);
            $this->load-
>view('supplier/password', $data);
            $this->load-
>view('supplier/footer', $data);
        }else {
            $this-
>Supplier_model->simpan_password();
        }
    }
}
/* End of file Supplier.php */
/* Location:
./application/controllers/Supplier.php */

```

Kisaran, 18 Juni 2025

Nomor : 131-318/PP-00.10/UR/VI/2025

Hal : Permohonan Riset

Kepada Yth,

**Pemilik Panglong Subur Makmur Kab. Batubara**  
di\_

Tempat

Dengan Hormat,

Semoga Tuhan Yang Maha Esa senantiasa memberikan kemudahan bagi kita semua dalam menjalankan aktivitas sehari-hari.

Universitas Royal sebagai salah satu perguruan tinggi penyelenggaraan program pendidikan Strata 1 (S1) dengan ini mengajukan permohonan kepada Bapak/Ibu agar kiranya dapat menerima Mahasiswa/I kami melaksanakan kegiatan Riset yang dilaksanakan pada tanggal **19 Juni 2025 - 19 Juli 2025**. Berikut ini data mahasiswa yang akan melaksanakan Riset :

No	Nama Mahasiswa	Program Studi	L/P	NIM
1	Irwan	Sistem Informasi-S1	L	21.22.0492

Adapun Tujuan dilaksanakannya Riset ini untuk Menyelesaikan Penulisan Skripsi Program Strata 1 (S1) Universitas Royal. Judul Skripsi : "*Implementasi Sistem SCM Untuk Mengoptimalkan Stok di Panglong Subur Makmur*".

Demikian permohonan ini kami sampaikan. Atas perhatian, kesediaan, dan kerjasama yang baik kami ucapkan terima kasih.

Hormat Kami  
Ditandatangani  
  
**Wani Mariatul Kifti, M.M.**  
NIDN. 0114057302

Tembusan :  
- Arsip

**PANGLONG SUBUR MAKMUR**  
**KABUPATEN BATU BARA**

Alamat : Dusun V Desa Lima-Laras Kecamatan Nibung Hangus

Lima-Laras, 20 Juni 2025

Kepada Yth.

**Pimpinan Universitas Royal Kisaran**

Di\_

Tempat

Dengan hormat,

Menanggapi Surat Permohonan Riset dari Universitas Royal Kisaran Nomor. 131-318/PP-00.10/UR/VI/2025 tentang izin Permohonan Riset kepada mahasiswa kami yang akan dilakukan pada tanggal **19 Juni 2025 - 19 Juli 2025**.

Adapun nama mahasiswa tersebut sebagai berikut:

No	Nama Mahasiswa	Progam Studi	L/P	NIM
1	Irwan	Sistem Informasi	L	21220492

Dengan tujuan dilaksanakan Riset untuk menyelesaikan Penulisan Skripsi Program Strata 1 (S1) Universitas Royal Kisaran dengan judul: **“Implementasi Sistem SCM Untuk Mengoptimalkan Stok di Panglong Subur Makmur”**.

Demikianlah Surat Pemberian Izin Riset ini dan atas perhatiannya kami ucapkan terimakasih.

Lima-Laras, 20 Juni 2025

Hormat kami

  
SUBUR MAKMUR  
Merjual :  
KAWA JENG B\* IN & ALAT BANGUN  
LISTRIK, TANI, DLL.  
DESA LIMA LARAS SIMP. TEMBOK  
TG.TIRAM-BATU BARA HP.0353 6167 335

**AMIRHAN**  
**PEMILIK**

  
SUBUR MAKMUR  
Merjual :  
KAWA JENG B\* IN & ALAT BANGUN  
LISTRIK, TANI, DLL.  
DESA LIMA LARAS SIMP. TEMBOK  
TG.TIRAM-BATU BARA HP.0353 6167 335



UNIVERSITAS ROYAL  
**FAKULTAS ILMU KOMPUTER**  
PROGRAM STUDI SISTEM INFORMASI

Office: Jln. Prof. H.M Yamin, SH No. 173 Telp & Faks. 0623-41079 Ext. 25 Lt. 1 Kisaran, Kab. Asahan, Prov. Sumatera Utara

**LEMBAR KONSULTASI SKRIPSI**

Nama Mahasiswa : IRWAN  
NIM : 21220492  
Program Studi : Sistem Informasi  
Judul Skripsi/TA : IMPLEMENTASI SISTEM SCM UNTUK MENGOPTIMALKAN STOK DI PANGLONG SUBUR MAKMUR  
Dosen Pembimbing : 1. RURI ASHARI DALIMUNTHE, M.Kom  
: 2. CHITRA LATIFFANI, M.Hum  
Lembar Konsultasi : Pembimbing 1  
Tanggal Pertemuan Pertama : 22 April 2025

PERTEMUAN/ TANGGAL	MATERI BAHASAN	PARAF DOSEN	KETERANGAN
22-04-2025	PERBAIKI BAB I  1. DOWNLOAD PANDUAN SKRIPSI 2. LATAR BELAKANG MENCAKUP (URGENSI, OBJEC PENELITIAN, PERMASALAHAN OBJEC, SOLUSI YG INGIN DIBERIKAN, SCM) 3. REFERENSI JURNAL TERBARU / BUKU MINIMAL 3 MENGGUNAKAN MENDELEY / TYPE IEEE DIATAS 2020 4. TAMBAHKAN DATA PENJUALAN PANGLONG 5. TENTUKAN JUMLAH DAN NAMA SUPLIER 6. SERTAKAN DAFTAR PUSTAKA		BAB I (Revisi)
14-05-2025	REVISI BAB I  1. PASTIKAN PAHAMI DATA 2. PAHAMI SUPLIER		BAB I (Revisi)
14-05-2025	ACC BAB I, LANJUT BAB BERIKUTNYA		BAB I (ACC)
22-05-2025	PERBAIKI BAB II  1. PAHAMI SCM 2. PAHAMI TINJAUAN PENELITIAN 3. PERBAIKI KERANGKA PEMIKIRAN 4. PERBAIKI SEJARAN INSTANSI/OBJEC PENELITIAN		BAB II (Revisi)
03-06-2025	ACC BAB II, LANJUT BAB III		BAB II (ACC)
03-06-2025	REVISI BAB III  1. PERBAIKI METODOLOGI PENELITIAN 2. PAHAMI KUALITATIF/KUANTITATIF 3. PERBAIKI NARASI OBERVASI SERTAKAN NAMA PEMILIK 4. PERBAIKI MAPS / TARIK RUTE KAMPUS KE TEMPAT PENELITIAN		BAB III (Revisi)
12-06-2025	ACC BAB III, LANJUT PERSIAPAN SEMPRO		BAB III (ACC)
24-06-2025	ACC SEMPRO  1. PAHAMI PROPOSAL		Seminar Proposal (ACC)

PERTEMUAN/ TANGGAL	MATERI BAHASAN	PARAF DOSEN	KETERANGAN
	2. PAHAMI OBJEK PENELITIAN 3. SIAPKAN DATA 4. SIAPKAN PPT		
09-09-2025	REVISI BAB IV 1. PERBAIKI SUB JUDUL PADA BAB IV, SESUAIKAN DENGAN PANDUAN 2. PAHAMI ALIRAN SISTEM INFORMASI		BAB IV (Revisi)
11-09-2025	ACC BAB IV 1. PAHAMI ALIRAN SISTEM INFORMASI 2. PAHAMI UML		BAB IV (ACC)
02-10-2025	REVISI BAB V 1. PERBAIKI SCRENSHOT SYSTEM 2. SESUAIKAN DENGAN ALUR		BAB V (Revisi)
10-10-2025	ACC BAB V, LANJUT BAB BERIKUTNYA		BAB V (ACC)
10-10-2025	ACC BAB VI, LANJUT PERSIAPAN SEMHAS		BAB VI (ACC)
10-10-2025	ACC SYSTEM 1. PAHAMI SYSTEM 2. PAHAMI SUPLIER, PAHAMI DATA 3. PAHAMI DATA LAPORAN		Pengujian Alat/Aplikasi (ACC)
10-10-2025	ACC SEMHAS 1. PAHAMI SKRIPSI 2. PAHAMI DATA 3. SIAPKAN DATA 4. LENGKAPI SKRIPSI 5. BELAJAR		Seminar Hasil (ACC)

Kisaran, 15 Oktober 2025

Ka. Prodi Sistem Informasi



**Nurwati, M.Kom.**  
NIDN: 0101068701



UNIVERSITAS ROYAL  
**FAKULTAS ILMU KOMPUTER**  
PROGRAM STUDI SISTEM INFORMASI

Office: Jln. Prof. H.M Yamin, SH No. 173 Telp & Faks. 0623-41079 Ext. 25 Lt. 1 Kisaran, Kab. Asahan, Prov. Sumatera Utara

**LEMBAR KONSULTASI SKRIPSI**

Nama Mahasiswa : IRWAN  
NIM : 21220492  
Program Studi : Sistem Informasi  
Judul Skripsi/TA : IMPLEMENTASI SISTEM SCM UNTUK MENGOPTIMALKAN STOK DI PANGLONG SUBUR MAKMUR  
Dosen Pembimbing : 1. RURI ASHARI DALIMUNTHE, M.Kom  
: 2. CHITRA LATIFFANI, M.Hum  
Lembar Konsultasi : Pembimbing 2  
Tanggal Pertemuan Pertama : 7 Mei 2025

PERTEMUAN/ TANGGAL	MATERI BAHASAN	PARAF DOSEN	KETERANGAN
07-05-2025	Lanjut Bab II		BAB I (ACC)
07-05-2025	Tambahkan daftar pustaka sesuai dengan judul penelitian dan perbaiki penulisan		BAB II (Revisi)
14-05-2025	Perbaiki penulisan dan lihat panduan skripsi		BAB I (Revisi)
18-06-2025	Lanjut Bab III		BAB II (ACC)
18-06-2025	Lanjut seminar proposal		BAB III (ACC)
25-06-2025	Lengkapi berkas dan kuasai materi		Seminar Proposal (ACC)
17-09-2025	Perbaiki penulisan sesuai dengan panduan penyusunan skripsi		BAB IV (Revisi)
09-10-2025	Lanjutkan Bab V		BAB IV (ACC)
09-10-2025	Perbaiki penulisan, sesuaikan dengan panduan penyusunan skripsi		BAB V (Revisi)
14-10-2025	Lanjutkan Pengujian Alat/Aplikasi		BAB VI (ACC)
14-10-2025	Lanjutkan Bab VI		BAB V (ACC)
14-10-2025			Pengujian Alat/Aplikasi (ACC)
14-10-2025	Persiapkan berkas dan kuasai materi		Seminar Hasil (ACC)

Kisaran, 15 Oktober 2025

Ka. Prodi Sistem Informasi



**Nurwati, M.Kom.**  
NIDN: 0101068701

## DAFTAR RIWAYAT HIDUP

### A. Identitas Pribadi

Nama Lengkap : Irwan  
Tempat/tanggal lahir : Lima-Laras, 16-03-2003  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Kebangsaan : Indonesia  
Suku : Jawa  
Alamat : lima-Laras Dusun VII kec.Nibung Hangus Kab.Batu Bara



### B. Riwayat Pendidikan

1. SDN 010170 Lima-Laras (2009-2015)
2. SMPN 3 Tanjung Tiram (2015-2018)
3. SMKN 1 Tanjung Tiram (2018-2021)
4. Universitas Royal (2021-2025)

Demikianlah daftar riwayat hidup ini diperbuat dengan sebenarnya.

Kisaaran, 19 September 2025  
Penulis,

**IRWAN**  
NIM: 21.22.0492